AUTHORING LARGE AND COMPLEX HYPERTEXT WITH REUSABLE COMPONENTS

A Dissertation

by

YUNG AH PARK

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

August 2010

Major Subject: Computer Science

Authoring Large and Complex Hypertext with Reusable Components

AUTHORING LARGE AND COMPLEX HYPERTEXT WITH REUSABLE COMPONENTS

A Dissertation

by

YUNG AH PARK

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

Approved by:

Chair of Committee,	Richard Furuta
Committee Members,	John Leggett
	Frank Shipman
	Ergun Akleman
Head of Department,	Valerie E. Taylor

August 2010

Major Subject: Computer Science

ABSTRACT

Authoring Large and Complex Hypertext with Reusable Components. (August 2010)

Yung Ah Park, B.S., Pohang University Science and Technology;

M.S., Oklahoma State University

Chair of Advisory Committee: Dr. Richard Furuta

caT, a Petri net-based hypertext system, supports the modeling of user characteristics, contextual information, as well as the policies that govern the operation of a digital library within the infrastructure that presents its contents. Traditionally, users have created caT networks from scratch, thus limiting their use to small collections. In this research, we introduce TcAT, a new authoring tool that supports features for component-based authoring, with a view to enable the creation of large caT nets that can represent complex, real-life spaces such as libraries and museums. TcAT supports graphical, template-based creation of nets as well as a textual language for easy manipulation of large structures. It implements composition operations from Petri net theory to select, categorize, and modify existing net fragments as building blocks for composing larger networks. Authors may switch modes between visual and textual authoring at will, thus combining the strengths of expressing large nets textually and selecting net fragments via point-and-click interaction. A user evaluation of the new authoring mechanisms suggests that this is a promising tool for improving the efficiency

of experienced users as well as that of novice users, who are unfamiliar with the Petri net formalism.

DEDICATION

To my parents, husband and daughter for their patience, encouragement, and love

ACKNOWLEDGEMENTS

I would like to express my deep gratitude to my committee chair, Dr. Richard Furuta. His valuable advice and guidance allowed me to persevere and finish my doctorial degree program. Without his patience and caring for me, I could not have completed my doctoral degree. I am also thankful to my committee members, Dr. John Leggett, Dr. Frank Shipman, and Dr. Ergun Akleman for their helpful suggestions and careful review of this work.

I thank my parents and parents-in-law for their love and endless support in so many ways. I also thank all my brothers and sisters and other family members.

I deeply give thanks to my husband, Dr. Eunjae Jung, for his love, belief, and encouragement. I am also thankful to my sweetheart, Young-Jee, who always brings happiness to me.

TABLE OF CONTENTS

CHAPTER Page

Page

LIST OF FIGURES

Page

Page

Page

Page

Page

LIST OF TABLES

Page

CHAPTER I

INTRODUCTION

In addition to serving as treasure troves of information artifacts, such as books, videos, and audio materials, libraries provide spaces for social interaction as well as specialized services for their patrons' diverse needs. While library staff may help patrons during the day, a limited query interface must serve their needs outside of office hours. Patrons may run into others who share their interests. Sometimes, books that they desire are unavailable, as other users have borrowed them. The environment of libraries is dynamic and vibrant with users who share the physical space. The dominant models of digital libraries trade the sense of space as well as the specialized services provided by skilled librarians for ubiquitous access to their materials. Contemporary digital libraries provide identical services to all patrons at all times, without regard to specific needs or special situations affecting these patrons.

context aware Trellis (caT) [Na 2001], a Petri-net-based hypertext system, lends itself particularly well to modeling the dynamism of traditional libraries, thus enriching the ubiquity of digital access with the sense of space, referral services, and other policies that govern the use of materials within libraries—both digital and physical. Digital libraries implemented in caT respond to characteristics of their patrons' physical and

---

This dissertation follows the style of *ACM Transactions on Information Systems.*

contextual environments. Users of caT-based libraries can interact with other users who are currently active, thus creating a sense of space. caT implementations include support for providing time-sensitive help options, such as live chat with a librarian during office hours or access to an automated help system after hours [Karadkar et al. 2002]. caT libraries also can tailor their content to suit the characteristics of a patrons' information device. For example, a patron reading a book or watching a video via a desktop computer connected to high speed Internet would be provided with a richer version than another, who may be viewing that book or video from a cell phone or via a dialup connection. caT hypertexts support these features by virtue of the properties that are inherent to Petri net structures without the need for specialized programming.

Due to a lack of support for sophisticated authoring techniques, authors of caT hypertexts have designed these from scratch. The effort in building a caT network from the ground up involves much effort in the initial design. Earlier versions of the authoring interface have supported visual authoring. While the visual interface allows authors to create hypertexts graphically, substantially large networks are difficult to design and present visually. The earlier attempt to ameliorate this situation included providing support for hierarchical authoring of nets [Jensen 1992] [Na 2001].

In this dissertation, we extend this approach by providing more advanced authoring mechanisms for large and complex hypertext documents that are often difficult to visualize, edit, and manage. We introduce a component-based authoring interface including advanced compositional mechanisms for composing Petri nets as building blocks to support authors in constructing large digital libraries that embody the

complexity of real-world libraries. This mechanism allows authors to select, categorize, and modify existing net fragments to suit new requirements for efficient management and reuse. We implement Petri net theory concepts, such as net transformation [Berthelot 1987] and Petri net algebra [Best et al. 2001] to help authors organize net elements into smaller units called component. A component net (CN) is a net fragment that consists of a set of places, transitions, arcs, and sub nets specified by authors and the system. We provide a predefined component net, called a template, for non-expert authors to build nets immediately and easily modify the nets by their needs.

Each component net stores its metadata such as name, description, functions, constraints, properties, summary, net type, media type, and structural pattern. Metadata is an important factor in characterizing and identifying components. This metadata helps identify the structure that is presented and not explicitly highlighted in a large specification.

caT displays a parent net and its subnets together with same level of abstraction. This causes difficulty in reading and understanding of large and complex specifications. For the efficient management and display of a large Petri net, we provide a configurable display at different levels of abstraction: for example, an abstract/reduced net view can be transformed to an expanded/refined net view and vice versa while preserving main properties/functionality. When Petri nets are displayed according to users' preferences, Petri nets are more readable and understandable. We provide easy creation, deletion and modification of subnets by using a tabbed panel mechanism. To reduce disorientation,

we also provide a hierarchical tree view that shows the overall structure of the hypertext. Through these features we can provide easy navigation among subnets and net elements

The current Motif-based authoring tool of caT (i.e., xTed) lacks an intuitive user interface. We reduce the complexity of the design processes and management of large and complex system by providing a more intuitive user interface for structured authoring. We provide more direct manipulation operations for copying net elements and selecting objects by using a mouse click and release. By using selection operations we can easily make a component and group objects. We provide easy manipulation of grouped objects with commands such as a move operation.

As other further enhancements of xTed's interface, we use tool-tips for instant visualization of attributes of objects. It is convenient to have a content layout window to display the content associated with a place on the canvas. Different kinds of views such as text editor and thumbnails are provided according to the author's preference. A document editing tool and other applications are invoked for easy creation and modification of the content that is associated with a place. Also we support zoom-in/zoom-out for dynamic rescaling of the net.

Even with its merits for small specifications, another hurdle in maintaining large nets is point-and-click interaction. Visual manipulation of net components is a slower process than is textual specification. The textual authoring language eases the manipulation of large nets by allowing authors to combine named templates by describing these connections textually. While individual places and transitions can be created using the textual language, it is most useful when recomposing a net from

existing net fragments. The textual specification allows authors to visualize large structures in a smaller space than is required for viewing the graphical structure of the net. It enables authors to recognize named net components easier, helping them find a relevant specification easily within a large collection of component net fragments. The textual language supports the semantics of the component net, albeit in a more readable form.

As a practical issue, motif-based xTed only executes under X windows. A system independent tool that executes in multiple platforms can increase the use of caT. To resolve this problem, we implement a proposed authoring tool called TcAT (Template-based caT Authoring Tool) using Java. TcAT is implemented using Java2D and Swing components. Also we enhance the implementation of the architecture of caT by changing the communication method to socket.

We conducted usability testing to verify usefulness of the proposed features. We present how much improvement is made when we compared with previous system and proposed prototype. Also, this testing discovered the significant characteristics for authoring large and complex hypertext. These discoveries would be helpful information for future implementations.

This dissertation is organized as follows. Chapter II describes literature reviews including the Petri net and its related theory and previous work on Petri net-based hypertext system such as Trellis and caT. Chapter III describes the system design of the proposed authoring interface focusing on our proposed authoring interface, which includes advanced composition, reuse, management, and display mechanism for easing

authoring of large and complex hypertexts. Chapter IV provides implementation details of our proposed system. Chapter V describes the usability evaluation of the proposed authoring prototype. Conclusions and future work are described in Chapter VI. We present the IRB document and answers from each subject in the usability evaluation in the appendices.

CHAPTER II

LITERATURE REVIEW

Composition of larger and more complex nets from existing nets has been extensively studied by many Petri net researchers; high-level nets, net transformations, Petri net algebra, modular nets and object nets, and so on. [Best et al. 2001] [Berthelot 1986]. This chapter reviews the literature about Petri net theory, as well as the related theories mentioned above.

As previous work to this dissertation, we provide description of Petri net-based hypertext systems such as Trellis and caT. Understanding these works is very important.

To generate the automatic display of a Petri net on the drawing panel of caT authoring tool, we studied and applied graph layout algorithms [Sugiyama et al. 1981] to the Petri net layout.

To generate the fragment library and find an appropriate library from a large repository, a pattern language [Alexander et al. 1977] can be used.

2.1 Petri Net

As shown in Figure 1, a Petri net, developed by Carl A. Petri in 1962, is a directed bi-partite graph that forms from the combination of transitions (represented in illustrations by rectangular boxes), places (circles), tokens (dots) and directed arcs (arrows). The formal definition of Petri net is defined as follows:

A Petri net structure is a 3 tuple, <P, T, F>, where:

- P is a finite set of places, $\{p_1, p_2, \ldots, p_n\}$ with $n \geq 0$;

- T is a finite set of transitions, $\{t_1, t_2, \ldots, t_m\}$ with $m \geq 0$
- F is the flow relation, $F \subseteq (P \times T) \cup (T \times P)$ a mapping representing arcs between places and transitions.

P and T are disjoint; ($P \cap T = \varnothing$). The preset of transition t ($\bullet t$) is the set of input place of t; $\bullet t = \{p \mid (p, t) \in F\}$. The postset of transition t ($t\bullet$) is the set of output place of t; $t\bullet = \{p \mid (t, p) \in F\}$.

A marking is a function that assigns a number of tokens to each place; $M: P \rightarrow N$, $N= \{0, 1, 2 \ldots\}$. A transition is enabled under marking M if there are enough tokens in each of its input places. Firing an enabled transition t in a marking M consumes one token from each of its input places and produces one token to each of its output places when the weight of each arc is 1. M_0 is the initial marking of the Petri net. The state of M is represented as a vector of tokens assigned to each place. The series of marking M ($M_0, M_1, M_2 \ldots M_f$) is generated from M_0 by an execution of the Petri net. Firing an enabled transition in a marking M invokes movement to a next state of M. For the Petri net in Figure 1, $M_0 = [0, 1, 0]$. Transition1 is enabled under Marking M_0. After firing Transition1, the next state of M becomes $M_1= [1, 0, 0]$.

When a Petri net that uses a basic token mechanism specifies complex systems, it leads to graphical complexity. For solving this problem, high-level Petri nets were developed by allowing the token to encode complex structured data. These include the Colored Petri net introduced by Jensen [Jensen1992]. The Colored Petri net is a four tuple $<\sum, P, T, F>$ that adds a finite set of token colors $\sum$ to the basic Petri net $< P, T, F>$.

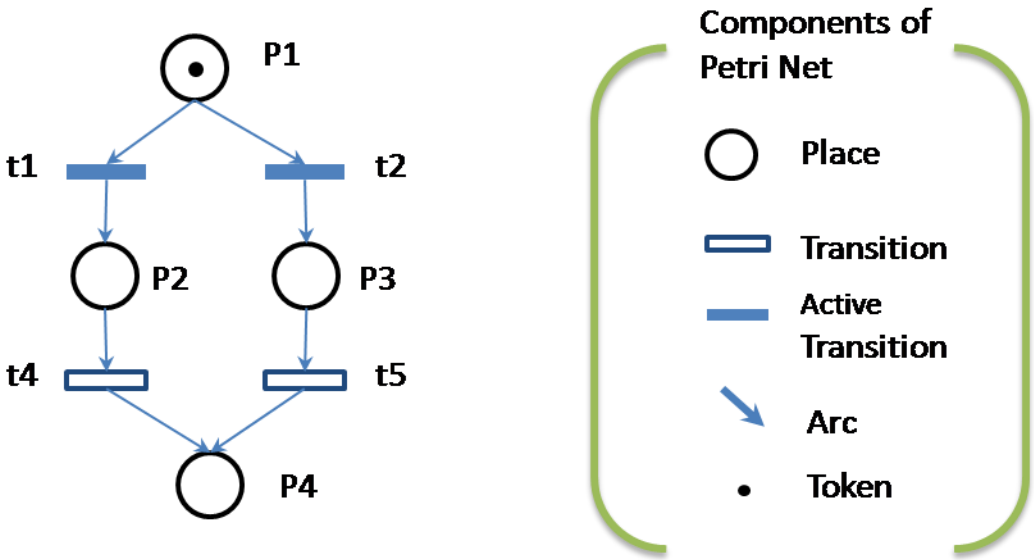


Figure 1: A Petri Net

2.1.1 Petri Net Algebra

Petri net algebra, called PBC (Petri net Box Calculus), combines process algebras like CCS (Calculus of Communicating Systems), COSY (Concurrent SYstems), and CSP (Communicating Sequential Processes) with Petri nets for description/specification and analysis of concurrent systems [Best et al. 2001]. The main objective of PBC is to support compositional Petri net semantics and an equivalent structured operational semantics (SOS). PBC is composed of a set of basic processes and a set of operators, which is from process algebra. CCS-like operators are formally defined for parallel, choice, sequential composition, synchronization, re-labeling, restriction, scooping and iteration. In PBC, Petri nets are treated as composable objects. Also, process algebra uses Petri net semantics for net-based verification. Proofs of algebraic properties of operators are provided in PBC.

2.1.2 Net Transformations

The definition of a set of transformations allows passing from one representation to another while keeping the main properties. Transformation modifies a net without affecting some of its interesting properties like deadlock freeness, 1-liveness, live-ness, and behavior equivalence. The goal of transformation is either to simplify or to refine a net, so every transformation is associated with a converse transformation. Reduction (or decomposition) techniques have been proposed to facilitate analysis by reducing a large-scale system model to a simpler one [Berthelot 1986] [Lee et al. 1987] [Pomello 1987].

In net transformation, composition is defined in terms of fusion of shared place and fusion of shared transition. Fusion of place represents synchronization by means of communication channels or common variables and fusion of transition represents synchronization by the rendezvous technique [Christensen and Petrucci 2000].

To refine Petri nets, researchers have used a top-down approach that provides conditions under which a transition or place can be substituted by a subnet [Suzuki and Murata 1983] [Zhou et al. 1989] and a bottom-up approach that constructs a large-scale system restrictively through an iterative integration of well-formed subnets [Chen et al. 1993] [Koh and Dicesare 1991].

2.1.3 Object Petri Nets

OPNs (Object Petri Nets) [Lakos 1995] [Lilius 2001] support an integration of object-oriented concepts into Petri nets, including inheritance and the associated polymorphism and dynamic binding.

To address the encapsulation of data and functions, subnet types as well as token types are included at a single class hierarchy of OPNs. OPNs support multiple levels of activity for modeling of complex systems at different levels of abstraction.

The subnet can be defined as a super (substitution) place or a super (substitution) transition and it makes that interaction between subnets can be either synchronous or asynchronous. A super place uses a transition fusion mechanism that provides interfacing a subnet with its environment via transitions and uses a synchronous channel for interaction between a super place and its neighboring transitions. A super transition uses a place fusion mechanism that provides interfacing a subnet with its environment via places and interacts with its environment through port places.

OPNs can be transformed to behaviorally equivalent Colored Petri Nets (CPNs) [Jensen 1992], thus we can adapt existing CPNs analysis techniques to OPNs. Implementations of OPNs in a textual language called LOOPN [Lakos and Keen 1991] and LOOPN++ [Lakos and Keen 1994] have been developed

2.2 Trellis and caT (context-aware Trellis)

Trellis is a Petri net-based hypertext model introduced by Furuta and Stotts in the late 1980's. By using Petri net structure and execution semantics, besides representing relationships among information fragments, Trellis specifies browsing semantics (i.e., "the manner in which the information is to be visited and presented") as part of the hypertext [Furuta and Stotts 1989, Stotts and Furuta 1989]. Hypertext in Trellis consists of the following components:

- Petri net representing linked structure of document
- Human-consumable components including content, windows, and buttons
- Projection mapping between Petri net and human-consumable components
- Projection mapping between logical elements of a hypertext and display mechanism

The formal definition of hypertext in Trellis is given [Stotts and Furuta 1989].

A hypertext H is a tuple H = <N, M_0, C, W, B, P_l, P_d > where:

- N is a Petri net structure, N= <P,T,F>;
- M_0 : an initial marking for N;
- C is a set of document contents;
- W is a set of windows;
- B is a set of buttons;
- P_l is a logical projection for the document;
- P_d is a display projection for the document

Text, graphics, table, bit maps, executable code, audio and another hypertext can be an element of C. The button, a symbolic name for a link, is an action that causes the current display to change in a specified way. P_l provides mappings from components of Petri net to human-consumable components of a hypertext H:

P_l = <C_l, W_l, B_l> where

- C_l: $P \rightarrow C \cup \{v\}$
- W_l: $P \rightarrow W \cup \{v\}$

- B_l: $T \rightarrow B \cup \{v\}$, v is symbol of a null value.

P_d is a collection of mappings between a button and a region on the screen with interaction devices such as a mouse button and between a window and proper physical display device for content.

A hypertext H with marking m, <H, m>, is the set of possible paths through a hypertext from a specific point. The current marking m decides which elements are displayed. When a certain place has a token, the contents mapped with the place are displayed. The set of elements displayed with marking M is

$$\{ C_l(p_i) \mid p_i \in P \text{ and } M(p_i) > 0 \}$$

The window of each input place of an enabled transition t_i, $\{W_l(p_j) \mid p_j \in I(t_i) \}$, displays button $B_l(t_i)$.

Trellis represents information content with “places” and links with “transitions”, expressing hypertexts within the formal definition of the Petri net structure. Directed arcs (arrows) indicate the browsing direction and colored tokens (dots) indicate the users’ current locations in the hypertext structure. A user corresponding to a token can access the contents of the place through a browser and may “fire” any enabled transitions to continue browsing to another place. A transition is enabled when all its input places have tokens that satisfy the activation condition of the transition. Only enabled transitions can be fired. When a transition is fired, one or more tokens from each input place of the transition are removed and one or more tokens are placed into each output place of the transition. Thus, a user may view the contents of several places simultaneously. Thus, Trellis separates the information content from the links (or actions) that are available to

the user. In contrast, Web pages embed links within the information content. Unlike Web documents, Trellis also separates the information content from its presentation; content elements are associated with places and browsers present this information with little or no direction from the server.

caT (context aware Trellis) extends Trellis to respond to characteristics of reader's physical and contextual environments. For this, features of high-level Petri net including structured tokens, flexible net description, and hierarchical nets are added. The formal definition of Petri net in caT is [Na 2001];

A caT Petri net structure is a tuple, HCPN = <S, ST, STM, IOM, I>, where:

- S = $<\Sigma, P, T, A, \tau, C, G, E>$ is a set of pages; each page, $s \in S$, is a nonhierarchical CPN= $<\Sigma_s, P_s, T_s, A_s, \tau_s, C_s, G_s, E_s>$;

  Σ, P, T, A, τ, C, G, and E represent all corresponding data (or functions) in S, a set of pages; Σ_s, P_s, T_s, A_s, τ_s, C_s, G_s, and E_s represent data (or functions) in each page s.

  - Σ_s is a finite set of token types called color sets;
  - $P_s = \{p_{s1}, p_{s2}, \ldots, p_{sn}\}$ is a finite set of places with $n \geq 0$;
  - $T_s = \{t_{s1}, t_{s2}, \ldots, t_{sm}\}$ is a finite set of transitions with $m \geq 0$ and $P_s \cap T_s = \varnothing$;
  - $A_s \subseteq (P_s \times T_s) \cup (T_s \times P_s)$ is the flow relation, a mapping representing arcs between places and transitions;
  - τ_s: $T_s \rightarrow \{0, 1, 2, \ldots, \infty\} \times \{0, 1, 2, \ldots, \infty\}$ is a function mapping each transition to a pair of values termed release time and maximum latency, respectively. For any transition $t \in T_s$, we write $\tau_s(t) = (\tau_s^r, \tau_s^m)$, and we

require that $\tau_s^r \leq \tau_s^m$;

- C_s: $P_s \rightarrow \Sigma_s$ is a color function mapping each place to a color set (type);
- G_s: $T_s \rightarrow$ *Boolean Expression* is a guard function mapping a Boolean expression to each transition;
- E_s: $(T_s \times P_s) \rightarrow$ *Arc Expression* is an arc function mapping an assignment expression to each output arc.

- $ST \subseteq T$ is a set of substitution transitions; T is a set of transitions in all pages;
- STM: $ST \rightarrow S$ is a substitution transition/page mapping function; no page is a subpage of itself;
- IOM: $ST \subseteq (P_{source} \times P_{target})$ is a input/output mapping function; P_{source} is a set of places (especially input or output places of ST transition) in the source page, and P_{target} is a set of places in the target page (STM(st));
- $I \in S$ is a start page.

The activation and firing conditions of a transition can be quite complex and are determined by rules written inside the transition and the arcs connected to it. Authors allow access to certain places or restrict it by setting conditions on arcs and transitions. Information may be displayed or hidden depending upon time, day of the week, physical location of the user, or whether the user is accessing this information for the first time. Browsing semantics are built into the structure of the document. Thus, caT is a flexible system, that separates the structure, content, presentation, and browsing semantics, each of which can be manipulated separately. Due to their expressiveness, caT hypertexts

grow and become more complex as authors express large information sets or describe behavior for several user groups under various circumstances.

Authors create caT nets using a graphical tool called xTed via point-and-click interaction and lay out net components on a two-dimensional canvas, as shown in Figure 1. The visual complexity of nets increases quickly as authors add more places and transitions to the net. To help mitigate this problem, xTed supports hierarchical nets [Na 2001]. This approach enables authors to create a network of layered sub nets and to deal with smaller net segments without being overwhelmed by the larger network. Conceptually, this approach helps authors deal with structurally or functionally similar components together. Visually, a hierarchical net replaces parts of a network with a simpler visual surrogate for the sub net that it represents. However, this support does not include the ability to combine or modify nets in a flexible manner. Typically, authors design networks with tens of nodes.

The Trellis/caT server, which runs the Petri net engine, sports no visual interface. Unlike Web servers, the Trellis/caT server is stateful and maintains the browsing details regarding all users. Users access the hypertexts through client programs called browsers. Trellis/caT browsers possess significant freedom in presenting the contents of the Trellis/caT network to the users. Different browsers can present a net and, indeed, a user's state of browsing differently. For example, while a Web browser like Internet Explorer can display the contents of a Trellis/caT net like a normal Web site, a text browser could display the textual content only, ignoring the images or audio. This allows users to access a single network from different viewpoints.

A hypertext's structure grows larger and more complex in response to the size of its content as also the variation of situations it supports. A hypertext has been designed to present different views of digital collections depending upon variables such as the time of access, user location, user preferences, and user privileges [Furuta and Na 2002]. MIDAS, an extension to caT, adapts the information display to suit the characteristics of the client device [Karadkar et al. 2004]. For example, users accessing the hypertext from an iPhone might see different details than users of desktop computers.

2.3 Pattern and Pattern Language

Patterns were originally introduced by Christopher Alexander [Alexander et al. 1977] who worked in architecture and urban planning. His idea has been applied to the object-oriented community and has become a hot-topic in the Software Engineering domain [Gamma et al. 1995] [Schmidt 1995]. Software patterns can help software developers resolve recurring problems that occur throughout software development. A pattern describes a recurring problem and its solution through expressing a relation between certain contexts. Name, problem, solution, and a description of consequence are the essential elements of a pattern [Gamma et al. 1995]. Patterns have been also applied within the Petri net community to define reusable Petri nets [Naedele and Janneck 1998] to enable the widespread use of Petri nets.

To generate the automatic display of a Petri net on the canvas of caT authoring tool, graph layout algorithms [Sugiyama et al. 1981] can be applied to Petri net layout algorithms.

2.4 Automatic Graph Drawing and Its Algorithms

It is not easy to draw a complex and large graph or even a small graph by human hand in order to allow good human cognition. Automatic graph drawing, where the computer is used to draw graphs automatically, has been widely studied since 1980 and many automatic graph drawing algorithms have been developed for several different types of graphs such as trees, directed graphs, planar undirected graphs, and compound graphs. Aesthetic criteria such as minimized crossing and minimized length of edge have been developed to characterize readability of graphs and to use to establish optimization goals of the graph drawing algorithm. Battista, et al. [Battista et al. 1994], provide a bibliographic survey of algorithms and Cruz and Tamassia [Cruz and Tamassia 1984] provide a graph drawing tutorial.

Hierarchical drawing for directed graphs [Sugiyama et al. 1981] [Carpano 1980] [Rowe et al. 1987] [Gansner et al. 1988] is one possibility that could be applied to Petri net drawing. The STT algorithm developed by Sugiyama, Tagawa and Toda [Sugiyama et al. 81] is a prominent algorithm reflecting this approach. A main aesthetic criterion of hierarchical drawing is minimized edge crossing. The divide and conquer method [Messinger 1988] partitions a large graph into sub-graphs, lays out the subgraphs with the STT algorithm, and then composes them.

2.5 Structural Analysis for Web Documentation

For structural analysis for Web documentation, Horie and Yamaguchi [Horie and Yamaguchi 2004] employed the non-well-founded set theory based on AFA (anti-foundation axiom). This set theory is used to represent link structures. To detect the

irregular structures of Web documents, such as incorrect and outdated links, they proposed reduction analysis that has three operations: arc detection, arc selection, and reduction. They assumed that regular Web pages can be translated into simpler AFA structures. That is, irregular links make the AFA structures more complex. If the removal of links makes the AFA structure simpler, these links might be irregular link candidates. The significance of an irregular link candidate can be evaluated based on the magnitude of unification resulting from reduction that is measured by hierarchical structure scheme and linear sequence scheme. As examples, they showed that obsolete pages and footnotes that occur rarely could be detected through this reduction.

CHAPTER III

SYSTEM DESIGN

Large, complex nets are often difficult to visualize, edit, and manage. To support authors in selecting and modifying net components for efficient management and reuse, we have developed an authoring interface, called the Template-based caT Authoring Tool (TcAT). TcAT implements Petri net theory concepts, such as net transformation [Berthelot 1986] and Petri net algebra [Best et al. 2001], to help authors organize net components into smaller units called templates. A template is a named collection of places and transitions that typically performs a specific operation.

3.1 Template-based Petri Net Composition

To create a net by composing existing nets as building blocks, we employ the component-based Petri net. A component net (CN) is a net fragment that consists of a set of places, transitions, arcs, and sub nets. The formal definition of component and component net is presented. Input and output ports (places) that connect to other components are defined in Single-Input/Single-output (SISO) form. A component net supports five operations, sequence (;), choice (+), parallel (||), iteration (μ), and refinement (r) to organize individual nets into larger structures [Berthelot 1986], [Best et al. 1995] [Fleischhack and Grahlmann 1998].

Formal Definition of Template-based Component Petri Nets

Component C = (Metadata$_j$, CN, subcomponent SCN$_i$)

$Metadata_j$: Metadata can be name, description, function (purpose), requirements (constraints), and property, structural pattern of CN, and so on. j is the number of metadata elements for CN, j>0, and i is the number of subcomponents.

Component net CN= $<\Sigma, P, T, F, \tau, C, G, E, I, O>$ [Hamadi and Benatallah 2003] [Murata 1989]

- $\sum$ is a finite set of token (color) types
- P is a finite set of places
- T is a finite set of transitions
- F is a set of directed arcs S, $F \subseteq (P \times T) \cup (T \times P)$
- I is input port (place); $\bullet I = \{x \in P \cup T \mid (x, I) \in F\} = \varnothing$
- O is output port (place); $O\bullet = \{x \in P \cup T \mid (O, x) \in F\} = \varnothing$ => There are no out coming arcs from O
- C is a color function mapping each place to a color type; $P \rightarrow \Sigma$
- E is an inscription of arc; $(T \times P) \rightarrow$ Arc expression
- G is a guard of transition; $T \rightarrow$ Boolean Expression
- τ is a function mapping each transition to time constraints (release time , maximum latency); $T \rightarrow \{0, 1, 2, \ldots, \infty\} \times \{0, 1, 2, \ldots, \infty\}$

Component has seven operations: sequence (;), choice (+), parallel (||), iteration (μ), refinement (r), union (U), and place fusion (F_P) :

$$C :: = \varepsilon \mid C ; C \mid C + C \mid C \parallel C \mid \mu C \mid r(C, a, C) \mid U(C, C) \mid F_P(C,C)$$

Definition: **sequence $C_1 ; C_2$** = (Metadata, CN, SCN_1, SCN_2) Where

Metadata = New Metadata

Structural Pattern = Sequence

SCN1 = C_1, SCN2 = C_2

CN = $<\Sigma, P, T, F, \tau, C, G, E, I, O>$ where:

$P = P_1 \cup P_2$

$T = T_1 \cup T_2 \cup \{t\}$

$F = F_1 \cup F_2 \cup \{(O_1, t), (t, I_2)\}$

$I = I_1$, $O = O_2$

$l = l_1 \cup l_2 \cup \{t,\tau\}$

Figure 2 shows the sequence operation for composing C_1 and C_2.

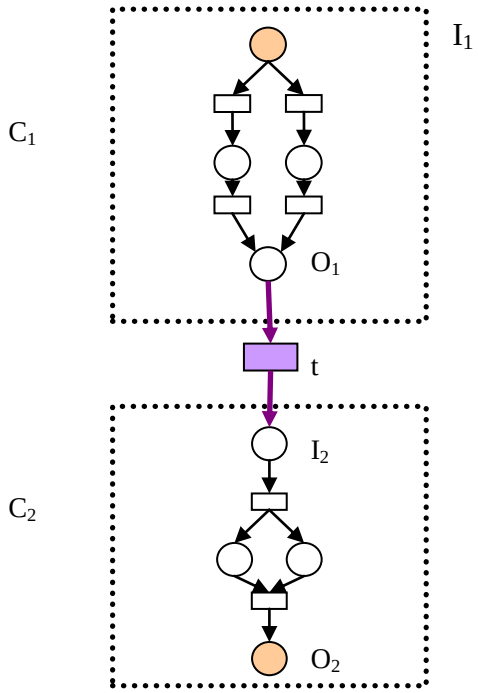


Figure 2: Sequence Composition of CN

Definition: **choice $C_1 + C_2$** = (Metadata, CN, SCN_1, SCN_2) Where

Metadata = New Metadata

Structural Pattern = Choice

$SCN_1 = C_1$, $SCN_2 = C_2$

CN = $<\Sigma, P, T, F, \tau, C, G, E, I, O>$ where:

$P = P_1 \cup P_2 \cup \{I, O\}$

$T = T_1 \cup T_2 \cup \{t_{I1}, t_{I2}, t_{O1}, t_{O2}\}$

$F = F_1 \cup F_2 \cup \{(I, t_{I1}), (I, t_{I2}), (t_{I1}, I_1), (t_{I2}, I_2), (O_1, t_{O1}), (O_2, t_{O2}), (t_{O1}, O), (t_{O2}, O)\}$

Figure 3 shows the choice operation for composing C_1 and C_2.

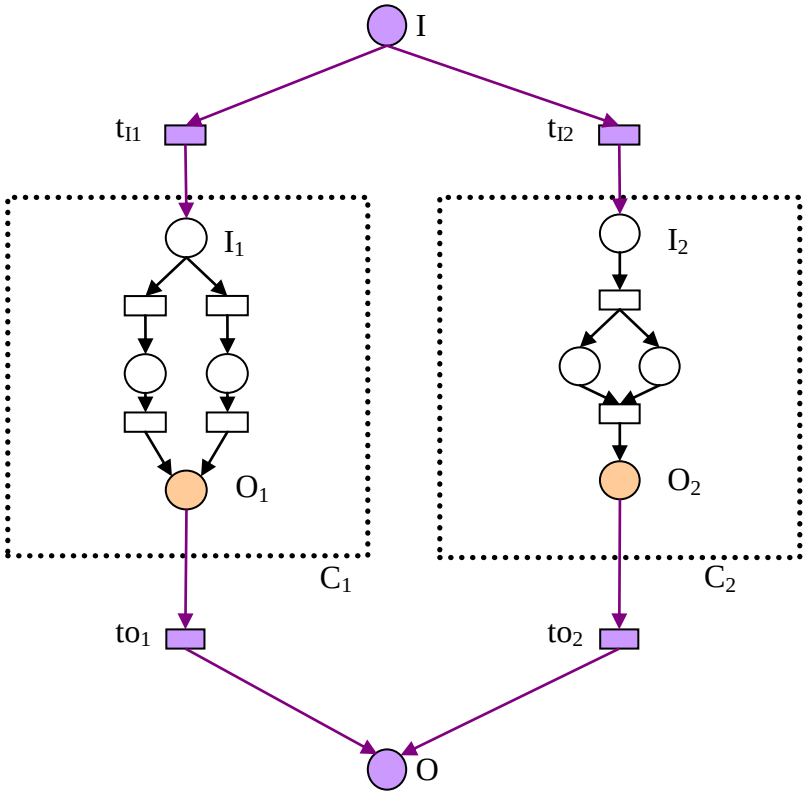


Figure 3: Choice Composition of CN

Definition: **parallel $C_1 \parallel C_2$** = (Metadata, CN, SCN1, SCN2) Where

Metadata = New Metadata

Structural Pattern = Parallel

$SCN_1 = C_1$, $SCN_2 = C_2$

CN =$<\Sigma, P, T, F, \tau, C, G, E, I, O>$ where:

$P = P_1 \cup P_2 \cup \{I, O\}$

$T = T_1 \cup T_2 \cup \{t_I, t_O\}$

$F = F_1 \cup F_2 \cup \{(I, t_I), (t_I, I_1), (t_I, I_2), (O_1, t_O), (O_2, t_O), (t_O, O)\}$

Figure 4 shows the parallel operation for composing C_1 and C_2.

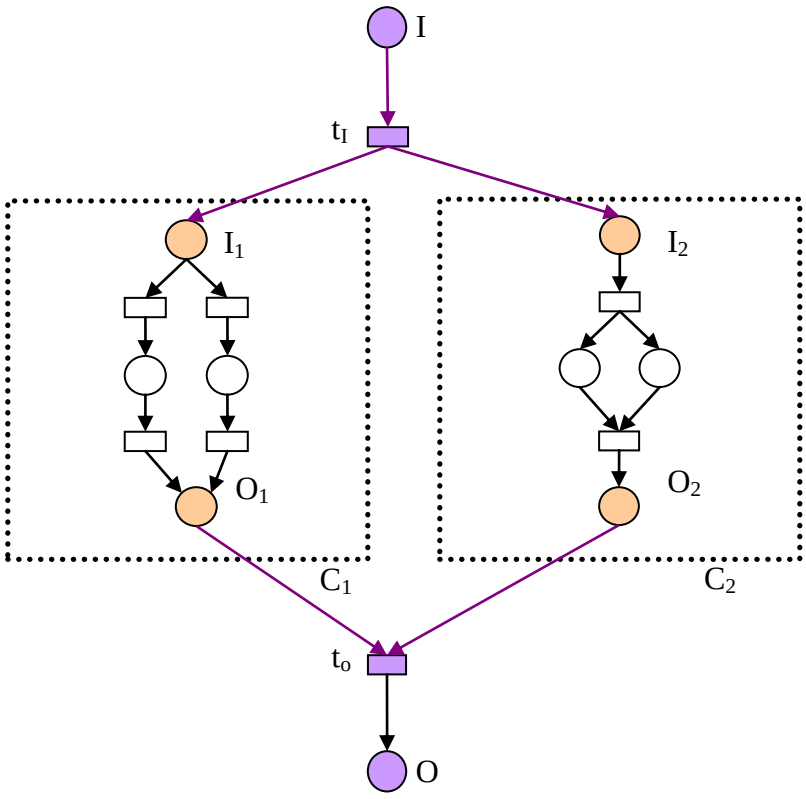


Figure 4: Parallel Composition of CN

Definition: **iteration μC_1**= (Metadata, CN, SCN_1) Where

Metadata = New Metadata

Structural Pattern = Iteration

$SCN_1 = C_1$

CN = $<\Sigma, P, T, F, \tau, C, G, E, I, O>$ where:

$P = P_1 \cup \{I, O\}$

$T = T_1 \cup \{t, t_I, t_O\}$

$F = F_1 \cup \{(I, t_I), (t_I, I_1), (O_1, t_O), (O_1, t), (t_O, O), (t, I_1)\}$

Figure 5 shows the iteration operation.

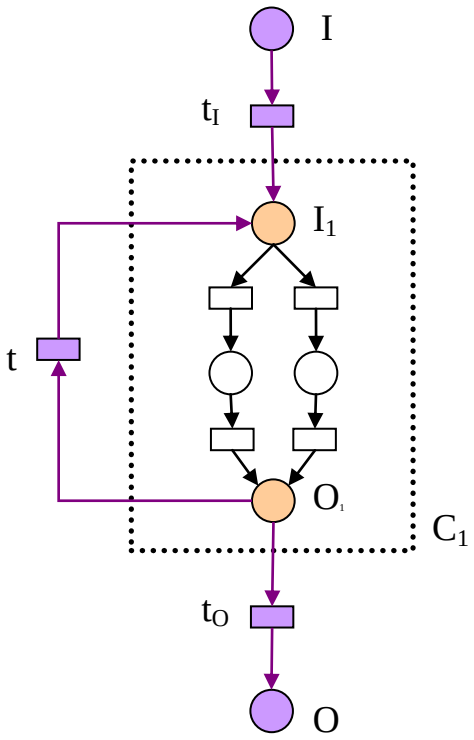


Figure 5: Iteration Composition of CN

Definition: **refinement r(C_1, a, C_2)** = (Metadata, CN, SCN_1, SCN_2) Where

Metadata = New Metadata

Structural Pattern = Refinement

$SCN_1 = C_1$, $SCN_2 = C_2$

CN = $<\Sigma, P, T, F, \tau, C, G, E, I, O>$ where:

$P = P_1 \cup P_2 - \{I_2, O_2\}$

$T = (T_1 - \{a\}) \cup T_2$,: "a" is the transition that can be substituted by other subnet.

$I = I_1$, $O = O_1$

Figure 6 shows the refinement operation that transition "a" at C_1 is substituted by a subnet C_2

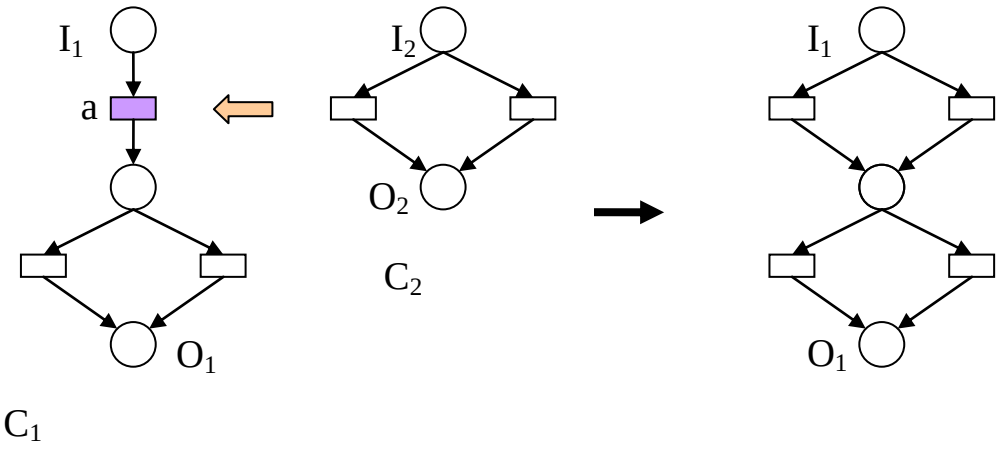


Figure 6: Refinement Composition of CN

Definition: **Place Fusion, $f_p(C_1, C_2)$** = (Metadata, CN, SCN_1, SCN_2) Where

Metadata = New Metadata

$SCN_1 = C_1$, $SCN_2 = C_2$

CN = $<\Sigma, P, T, F, \tau, C, G, E, I, O>$ where:

$P = P_1 \cup P_2 - \{I2\}$

$T = T_1 \cup T_2$

$F = ((F_1 \cup F_2) - \{(I_{2,}, t_k)\}) \cup (O_1, t_k)$;

$t_k \in I_2\bullet = \{x \in P_2 \cup T_2 \mid (I_2, x) \in F_2\}$

$I = I_1$, $O = O_2$

Figure 7 shows the combining procedure for C_1 and C_2 via the place fusion operation.

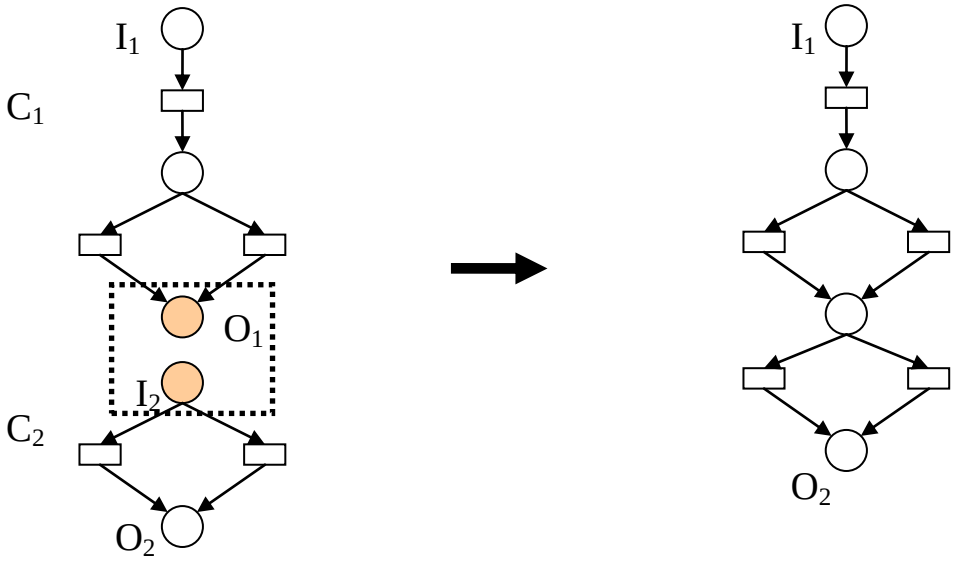


Figure 7: Place Fusion Refinement Composition of CN

Definition: **Union U(C_1,C_2)** = (Metadata, CN, SCN_1, SCN_2) Where

Metadata = New Metadata

$SCN_1 = C_1$, $SCN_2 = C_2$

CN = $<\Sigma, P, T, F, \tau, C, G, E, I, O>$ where:

$P = ((P_1 \cup P_2) - \{ I_1, I_2, O_1, O_2 \}) \cup I \cup O$

$T = ((T_1 \cup T_2) - (I_1\bullet \cup I_2\bullet \cup \bullet O_1 \cup \bullet O_2)) \cup I\bullet \cup \bullet O$

$F= ((F_1 \cup F_2) - \{(I_1, t_{Ik}), (I_2, t_{2k}), (t_{1m},O_1), (t_{2m},O_2,)\}) \cup \{(I, t_I) ,(t_O,O)\}$

$I=\text{Merge}(I_1, I_2)$, $O=\text{Merge}(O_1, O_2)$

$I\bullet = \text{Merge}(I_1\bullet, I_2\bullet)$, $\bullet O = \text{Merge}(\bullet O_1, \bullet O_2)$,

t_{ik} is transformed into t_I, t_{im} is transformed into t_O

$t_{ik} \in I_i\bullet = \{x \in P_i \cup T_i \mid (I_i, x) \in F_i\}$,

$t_I \in I\bullet = \{x \in P \cup T \mid (I, x) \in F\}$

$t_{im} \in \bullet O_i = \{x \in P_i \cup T_i \mid (x, O_i) \in F_i\}$

$t_O \in \bullet O = \{x \in P \cup T \mid (x, O) \in F\}$

Figure 8 shows the combining procedure for C_1 and C_2 via the union operation.

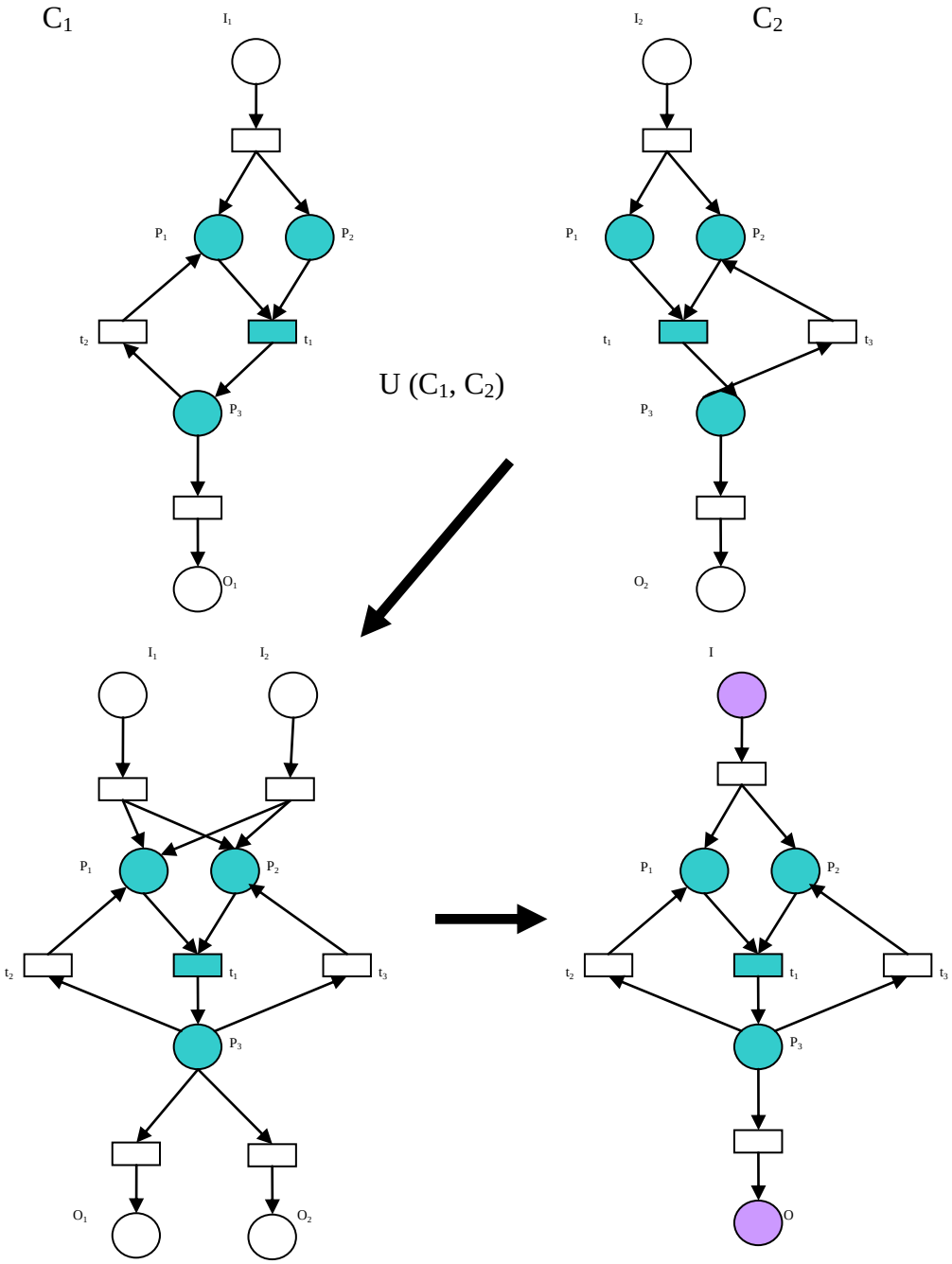


Figure 8: Union Transformation of CN

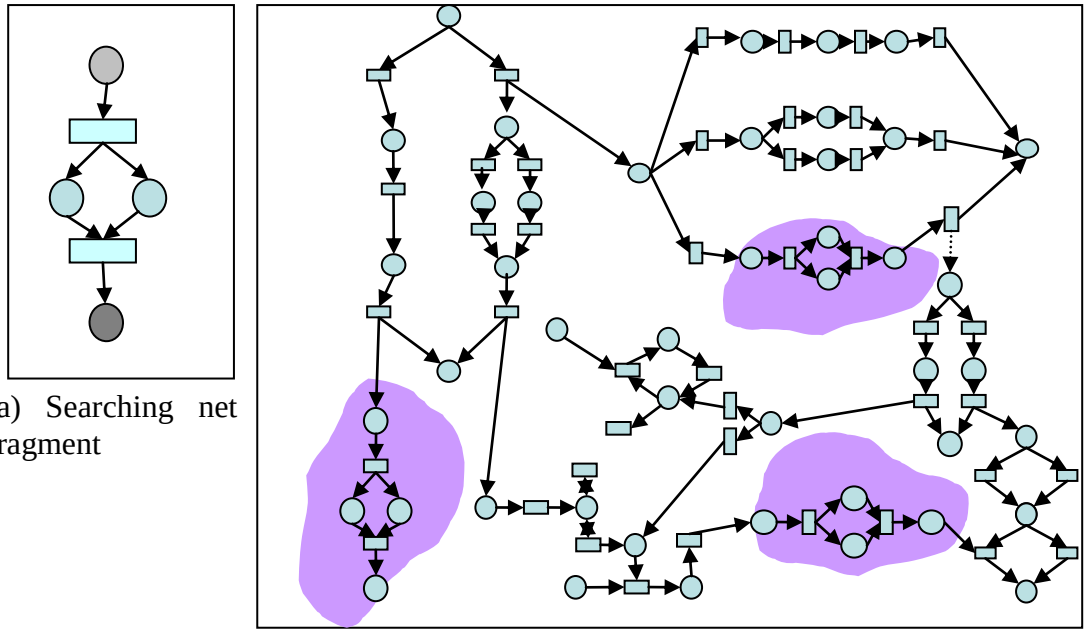
For the sequence operation, a new transition is required to link from the output place of the previous net to the input place of the next net. A new input place, a new output place, four new transitions (two transitions from the new input place to each subnet's input, and another two transitions from each subnet's output place to the new output place) are automatically generated for a new composed net by choice. For parallel operation of two subnets, a new input, a new output place, and two new transitions are generated; one is from the new input to both subnets' inputs and the other is from both subnets' outputs to the new output. When both subnets finish execution, the last transition can be activated.

Composing operations enable authors to construct larger hypertexts from existing fragments. For example, when an author creates a combined digital collection for fine artists, it can be composed from individual collections of Vincent van Gogh (named GL) and Claude Monet (i.e., ML). If an author wants to append Monet's collection to van Gogh's, she would use sequence operation (GL ; ML), sparing her the work of creating a new hypertext of the combined works. Instead if she desires to provide her users with an option of browsing either collection, she would use the choice operation (GL + ML). To present the two collections concurrently, presumably to help users compare and contrast the styles of these painters, she would use the parallel operation (GL || ML). The iteration option requires the browsing of the net fragment a certain number of times. The refine operation replaces a transition with a subnet, similar to specification of caT's hierarchical composition mechanism [Jensen 1992] [Na and Furuta 2001]. For each of

these operations, TcAT encloses the individual collections within control structures that generate the desired effect.

Authors use net transformation to modify (refine or simplify) a net structure. Net transformation is used to analyze a large and complex net by replacing an equivalent simplified representation with identical properties. We have implemented two transformations: union, which combines two structures with a shared substructure, and fusion, which combines two copies of a substructure within one structure [Berthelot 1986].

Each component net stores its metadata such as name, description, functions, constraints, properties, summary, net type, media type, and structural pattern. A net's metadata is an important factor in characterizing and identifying components. Initially, the author provides metadata manually. When we import existing nets to construct a CN, the CN automatically inherits some metadata from the existing nets. Using metadata, the structure that is presented and not explicitly highlighted in a large specification can be identified as shown in Figure 9. Figure 9 (a) shows the net fragment, which is a basic parallel structure that the user wants to find and 9 (b) shows the net fragment as identified in the specification. The found net can be substituted by a structurally equivalent net.



(a) Searching net fragment

(b) Identifying the relevant net fragments

Figure 9: Identification of a Specific Fragment

A template is a pre-defined component net. We provide some templates as a part of the system. Users retain the freedom to design and save additional templates. Using the pre-defined templates, non-expert authors can build nets quickly and easily modify the nets to suit their needs. For example, if authors want to add help to a certain net fragment, they employ a help template. Figure 10 shows the help template applied to net N_1. An author wishes to provide help regarding the contents of place P_1 in net N_1. By applying a union net transformation at P_1 in N_1 and P_1 at the help template, two copies of P_1 are combined into one place. The net N_1 and the help template are now combined into a single structure. In a large net, this transformation can be applied either to a single instance of N_1 or to all net fragments of this form.

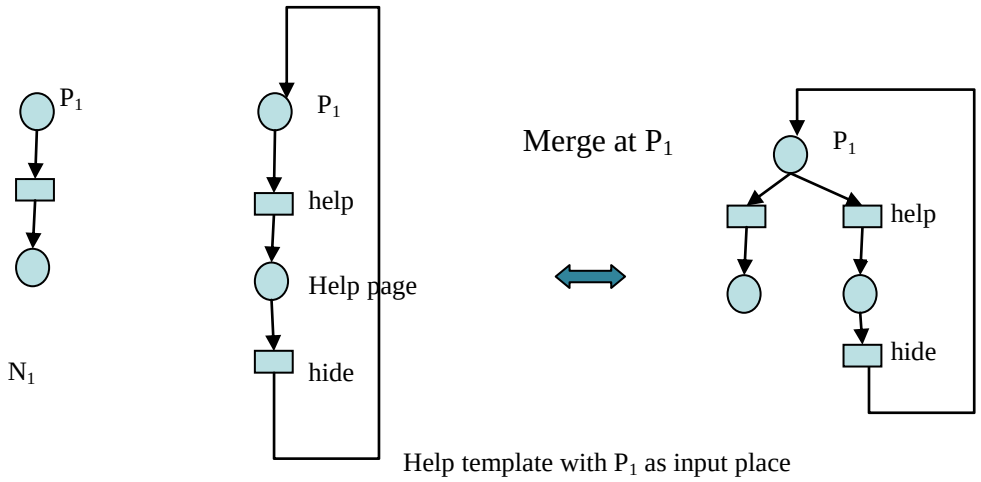


Figure 10: Net N_1 and its Modification Adding Help Template

Figure 11 illustrates a reduction rule [Lee et al. 1987] [Tu et al. 1990] in net transformation. In this case, the reduction rule removes one of two parallel nodes, which have identical input and output transitions. P_1 and P_2 are parallel nodes since they lead to the same set of input and output places via transitions t_1, t_2, and t_3. One of these nodes, along with its incoming and outgoing arcs, can be removed without affecting the semantic structure of the net that contains this fragment.

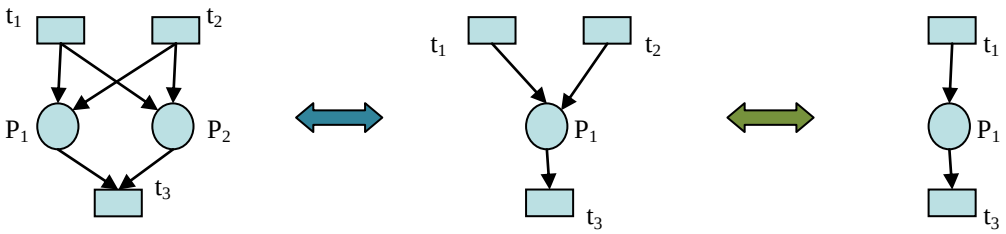


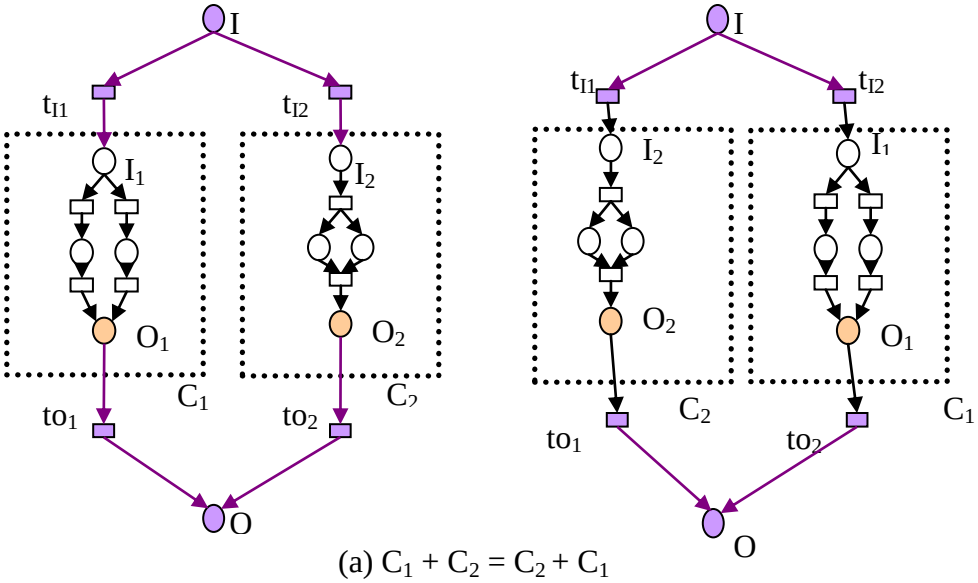
Figure 11: Removal of a Parallel Redundant Node

We cannot directly apply this reduction net transformation to dynamic caT structure adaptation. Even though P_1 and P_2 are the same structure, P_1 and P_2 may have different content. For nodes to be considered as parallel, P_1 and P_2 must have the same information content as well. For example, if P_1 and P_2 have same content but different media types (e.g., P_1 contains text and P_2 contains audio) these may be considered as parallel nodes as a browser can convert between the two forms automatically. We identify parallel places using place metadata. Parallel nodes can also be added to support different media types when browsers cannot convert content automatically between these types.

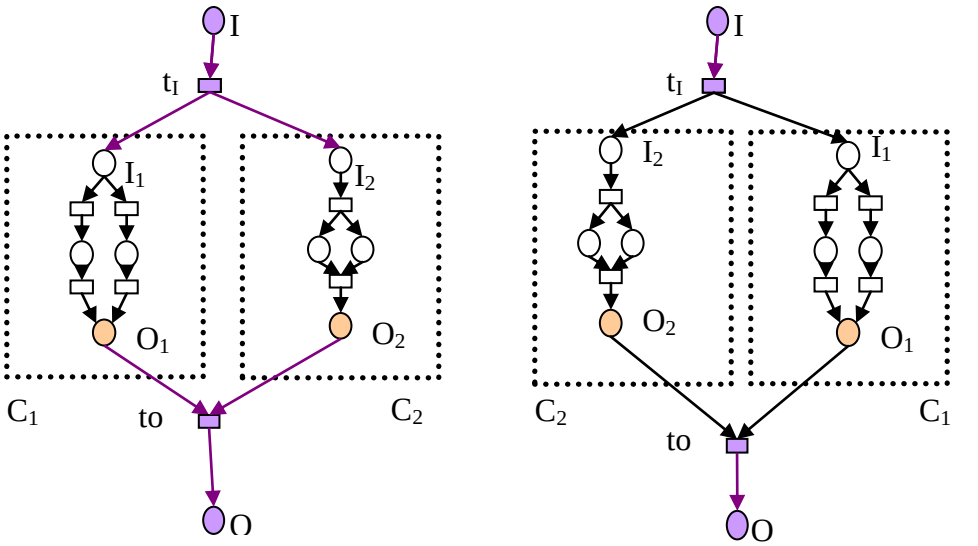
Commutative and associative properties involve both choice and parallel operation as presented in Table 1, Figure 12, Figure 13, and Figure 14.

Table 1: Algebraic Property for Composition Operation

Property	**choice**	**parallel**
commutative	$C_1 + C_2 = C_2 + C_1$	$C_1 \parallel C_2 = C_2 \parallel C_1$
associative	$(C_1+C_2)+C_3 = C_1+(C_2+C_3)$ $= C_1+C_2+C_3$	$(C_1\|\|C_2)\|\|C_3 = C_1\|\|(C_2\|\|C_3)$ $= C_1\|\|C_2\|\|C_3$

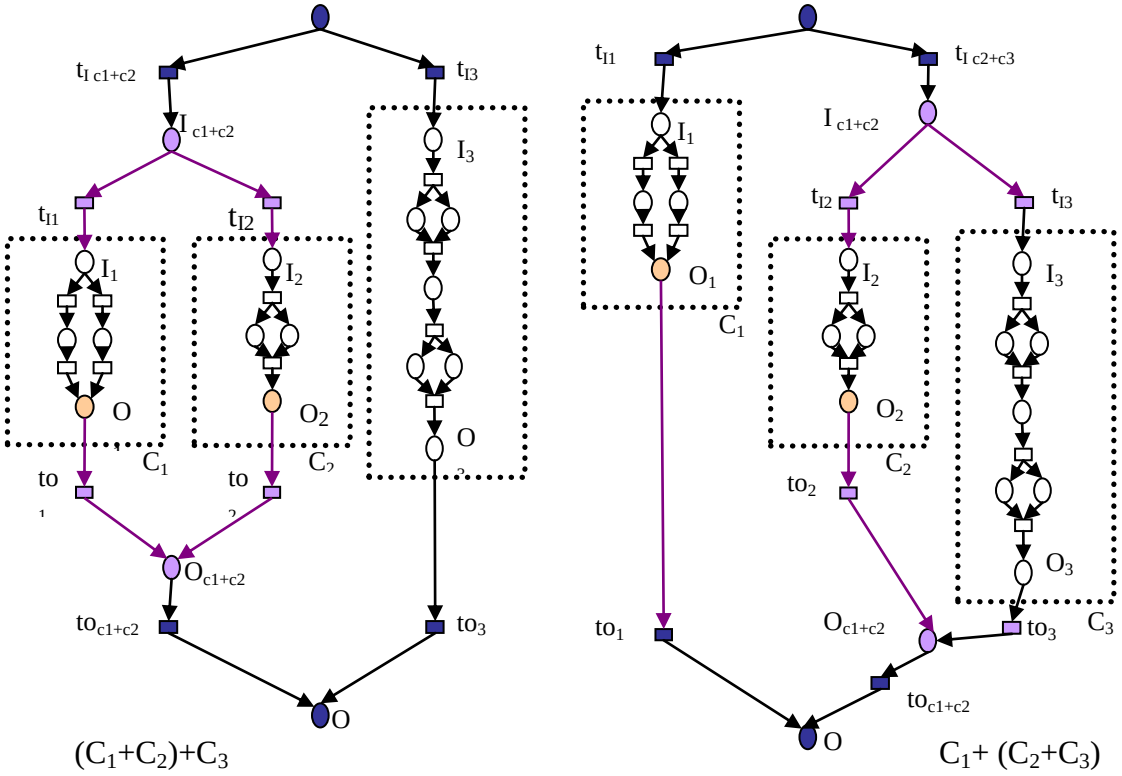


(a) $C_1 + C_2 = C_2 + C_1$



(b) $C_1 \parallel C_2 = C_2 \parallel C_1$

Figure 12: Commutative Property



$$(C_1+C_2)+C_3 = C_1+(C_2+C_3) = C_1+C_2+C_3$$

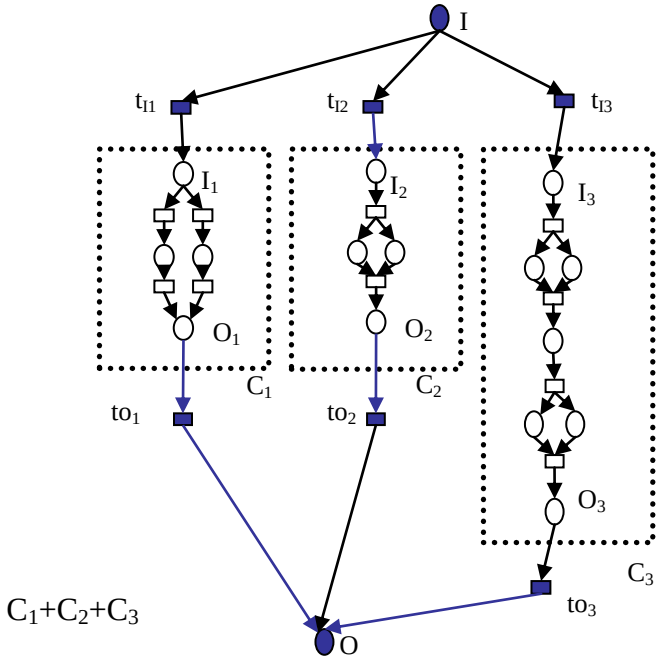
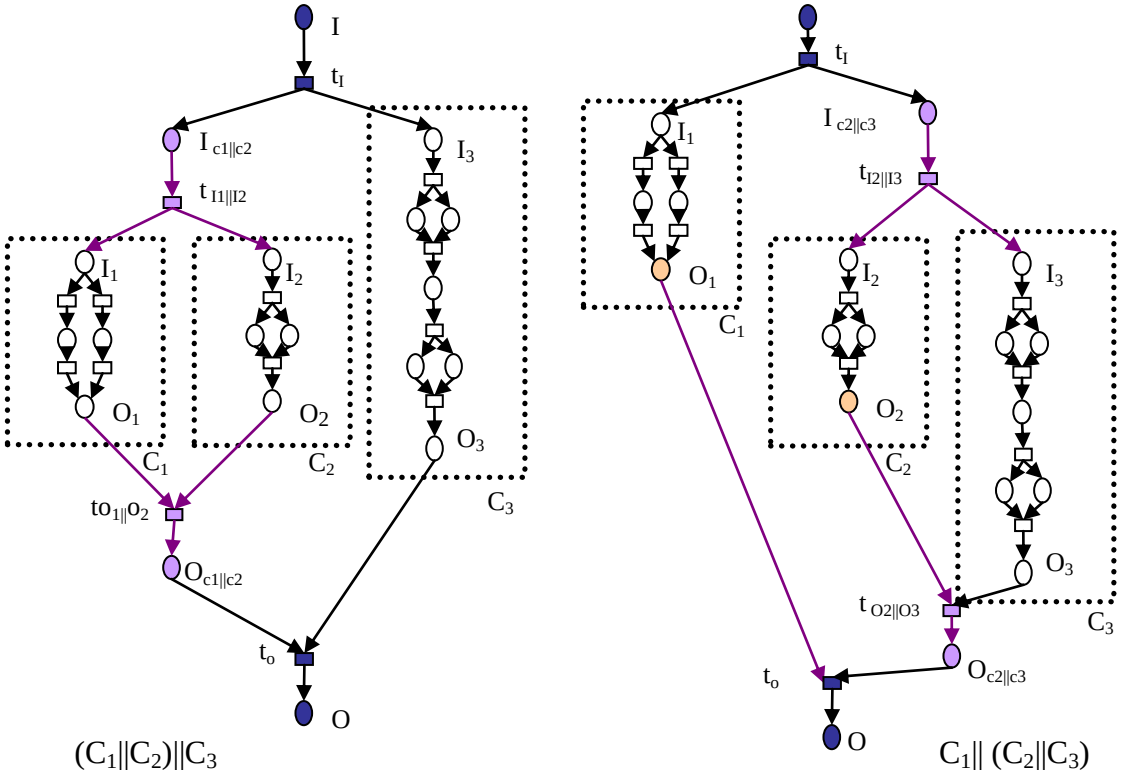


Figure 13: Associative Property for Choice Composition



$(C_1||C_2)||C_3 = C_1||\ (C_2||C_3) = C_1||C_2||C_3$

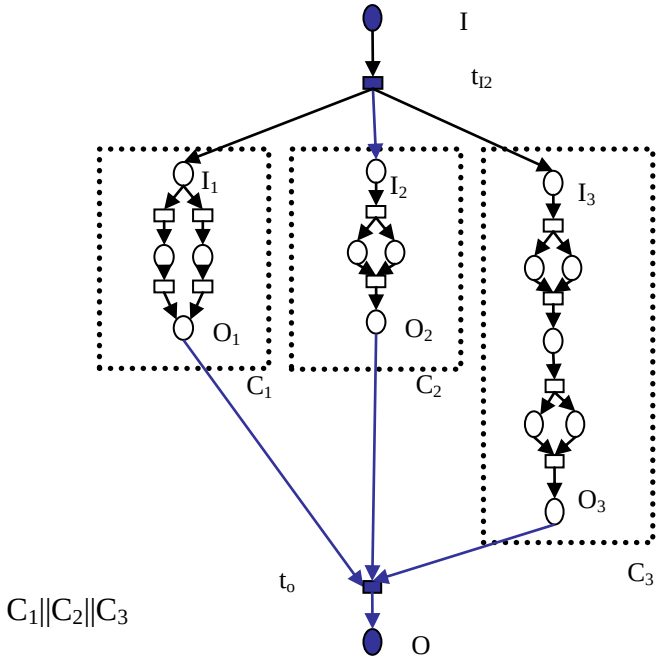


Figure 14: Associative Property for Parallel Composition

3.2 Textual Authoring Language

Another hurdle in maintaining large nets is the point-and-click interaction. Visual manipulation of net components is a slower process than is textual specification. The textual authoring language eases the manipulation of large nets by allowing authors to combine named templates by describing these connections textually. TcAT's text editor doubles as a textual authoring environment for caT nets. TcAT converts the textual specification to the graphical form automatically. Figure 15 shows the textual language and its graphical layout. While individual places and transitions can be created using the textual language, it is most useful when recomposing a net from existing net fragments. The textual specification allows authors to visualize large structures in a smaller space than is required for viewing the graphical structure of the net. It enables authors to recognize named net components easier, helping them find a relevant specification easily within a large collection of component net fragments.

The textual language uses the semantics of the component net. For example, C1;C2 is represented as sequence (C1, C2) and C1+C2 as choice (C1, C2). A text editor is integrated into the authoring tool. We provide two levels of specification: element-level specification and composition-level specification.

```
CompositionN3 {
   subNet N1;
   subNet N2;
   Net N3;
   N3=sequence (N1, choice (N1, N2));
}
Element N1  {
   Place p1, Place p2, Transition t1 ;
   Arc(p1,t1) ; Arc(t1, p2) ;
}
```

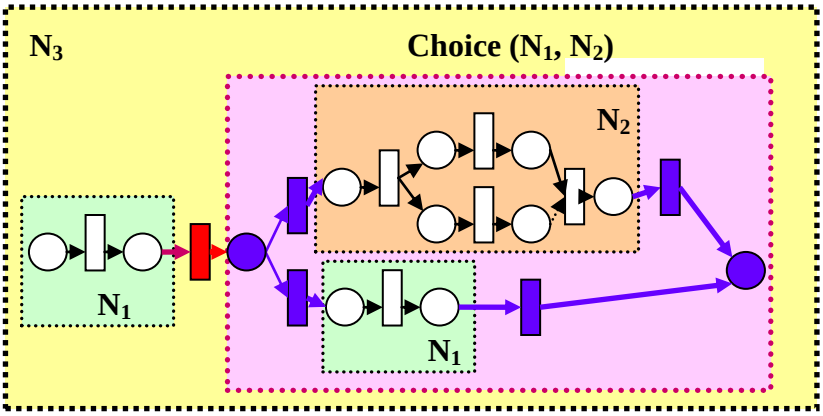


Figure 15: Textual Authoring Language Illustration

3.3 Template-based caT Authoring Tool

TcAT is implemented using Java2D and Swing components. Net components are implemented as Jcomponent elements. The TcAT interface, as shown in Figure 16, consists of a drawing panel, a tree view-based navigation window, content layout, textual editor, template panel, search window, and a document editor. Figure 17 presents login dialogue when TcAT starts.

Along with using provided templates, authors can customize the template. The template panel shows the current available templates as presented in Figure 18.

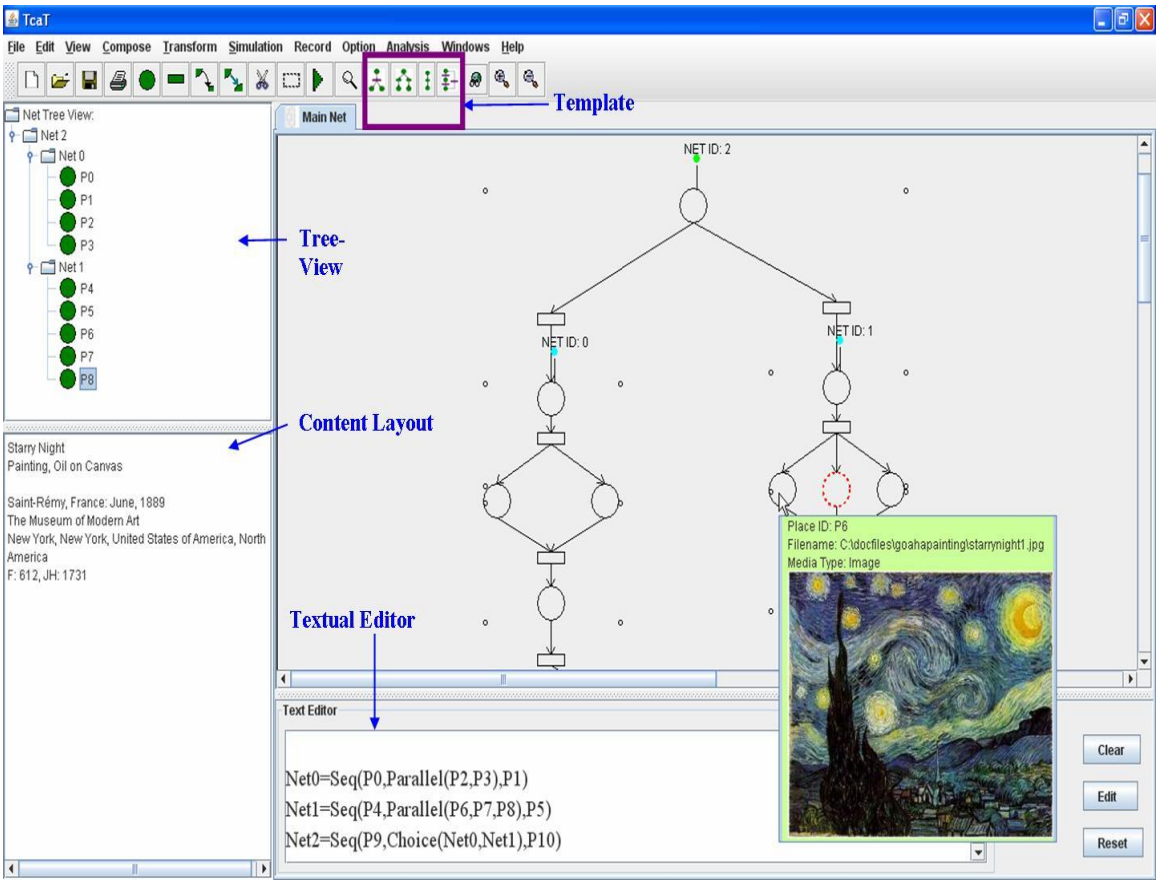


Figure 16: Template-based caT Authoring Tool (TcAT)

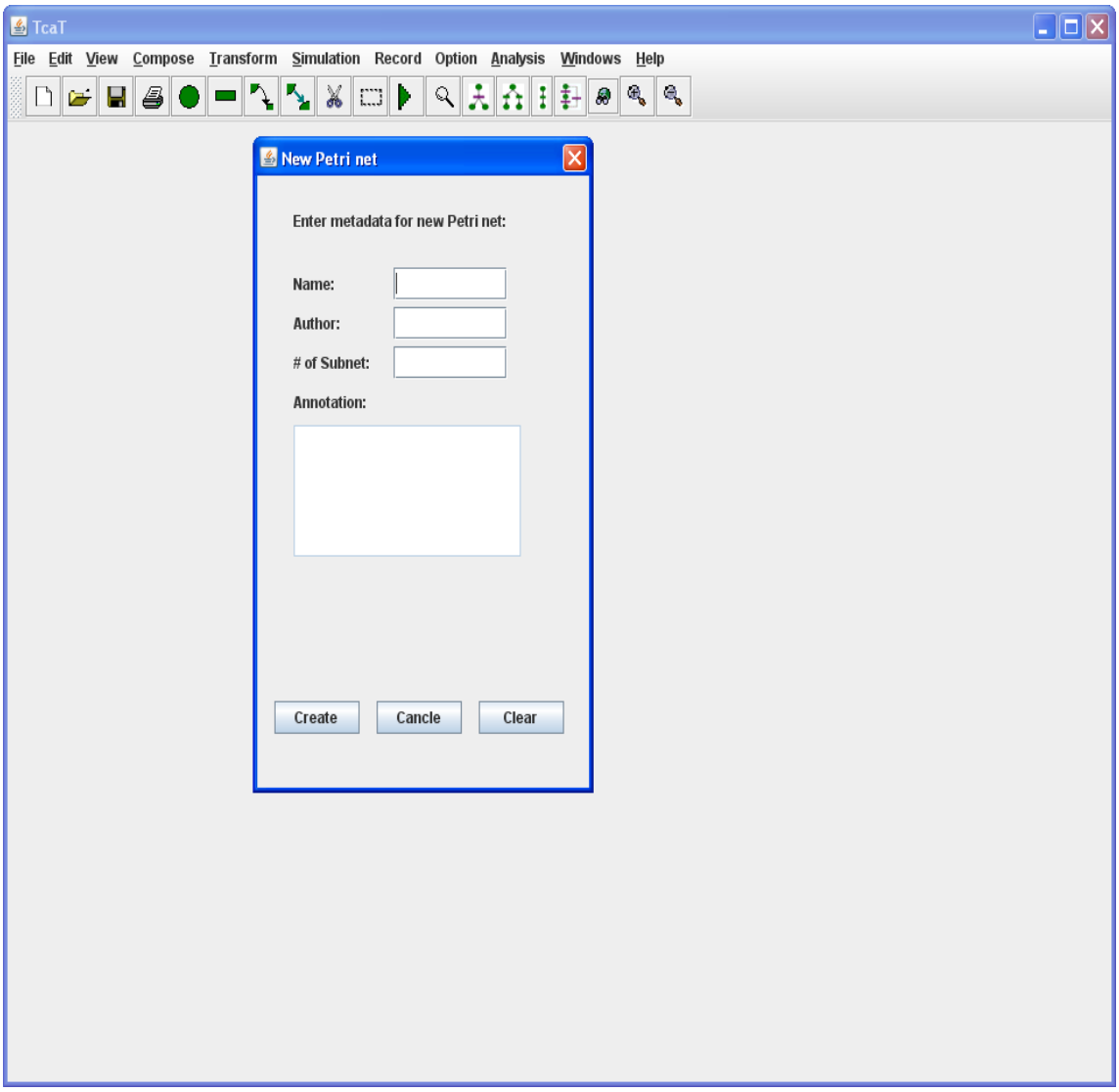


Figure 17: Login Dialogue

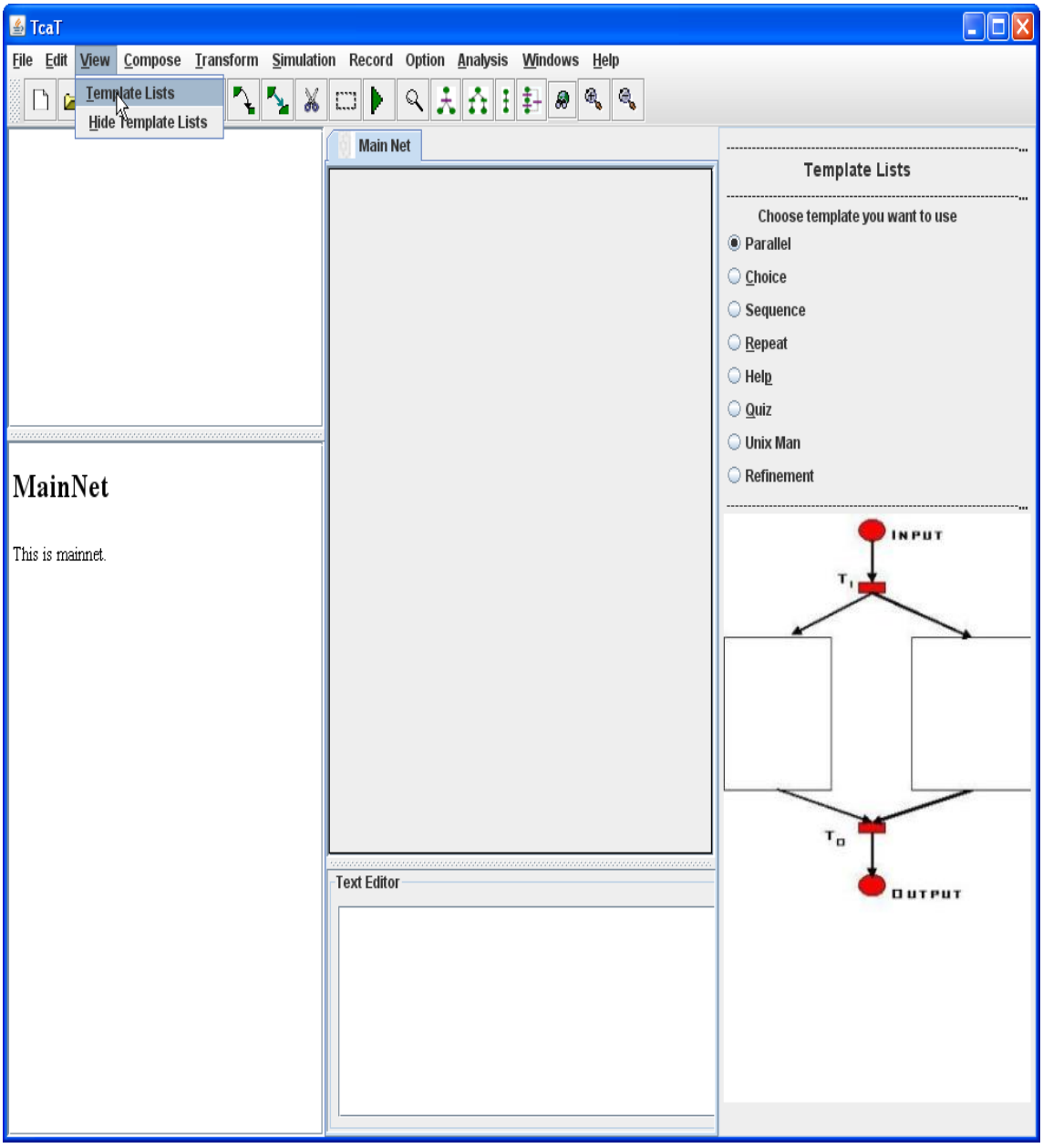


Figure 18: TcAT with Template Panel

3.3.1 Tree View-based Navigation

The tree view helps authors visualize the overall net structure for convenient navigation among net components. A content layout window displays associated content with a place on the canvas as presented in Figure 19.

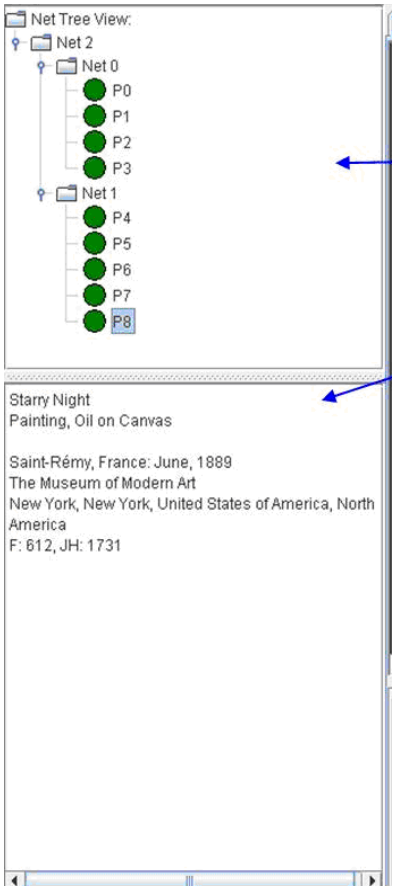


Figure 19: Tree View Navigation and Content Layout

3.3.2 Tool Tips

Instant views of the content for a place, author added metadata and annotations for net element are displayed as tool tips as will be illustrated later in this chapter. Tool tips also display thumbnail view of the captured structure of each net along with net

metadata, annotation and the net's textual language at the input place of a net. For a collapsed net, a tool tip displays a thumbnail of original structure of the net at the input place (representative node). This helps authors get a quick preview of the structure of collapsed nets.

3.3.3 Automatic Net Creation

TcAT provides the template net for automatic net creation. Basically, we provide four library nets that are the basic structural Petri net types --sequence, choice, parallel, and repeat-- as shown in Figure 20. After choosing a type of net, an author enters data for net attributes that can be metadata for the net, as presented in Figure 21.

Figure 20: Library Net

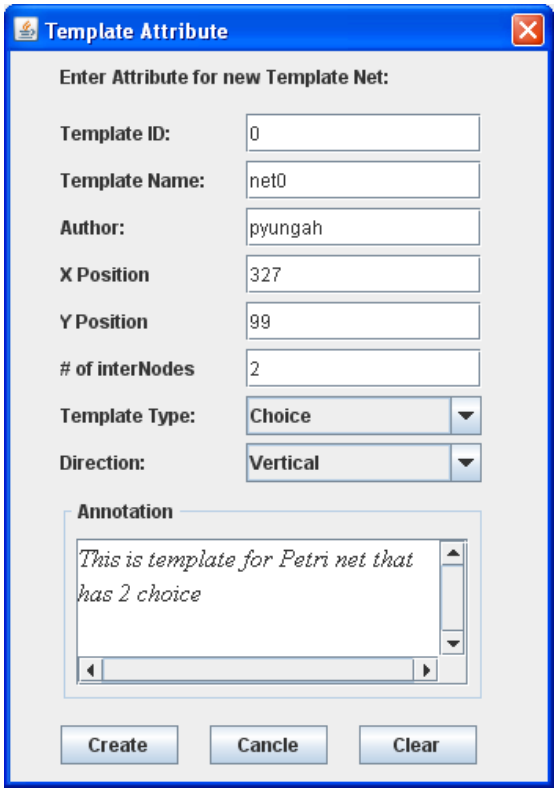


Figure 21: Net Attributes

Each field in Figure 21, except "# of interNodes" and "Template Type", has its own default value automatically assigned when an author does not provide information. At first Template ID is assigned to 0. The next ID is incremented by 1 from the last registered ID. The template name is made by concatenating "net" and ID (e.g., net0). Login name hands over to "Author" field. The position of x and y is from the position mouse clicked on. Vertical is a default value of "Direction" field.

Figure 22 presents the "net0" that is automatically created by library net. When net0 is displayed on the drawing canvas, TcAT automatically creates a tree and textual specification for net0. Figure 23 shows the tool tip displaying the net's information including a thumbnail of the captured structure of net0.

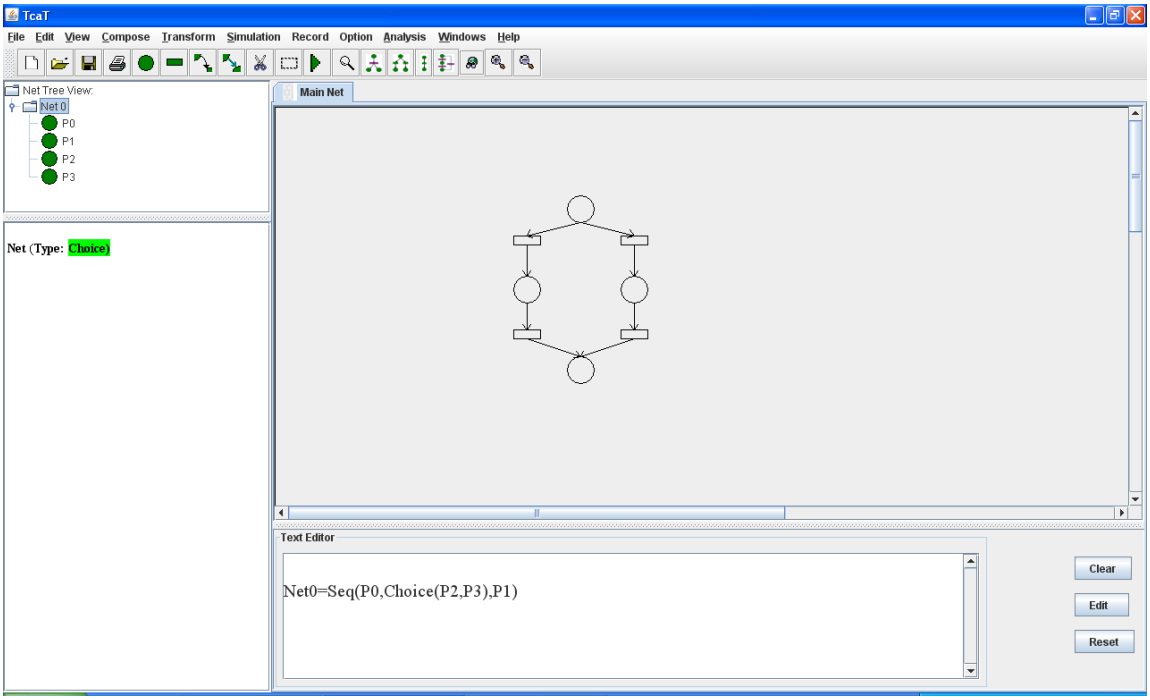


Figure 22: Automatic Net Creation

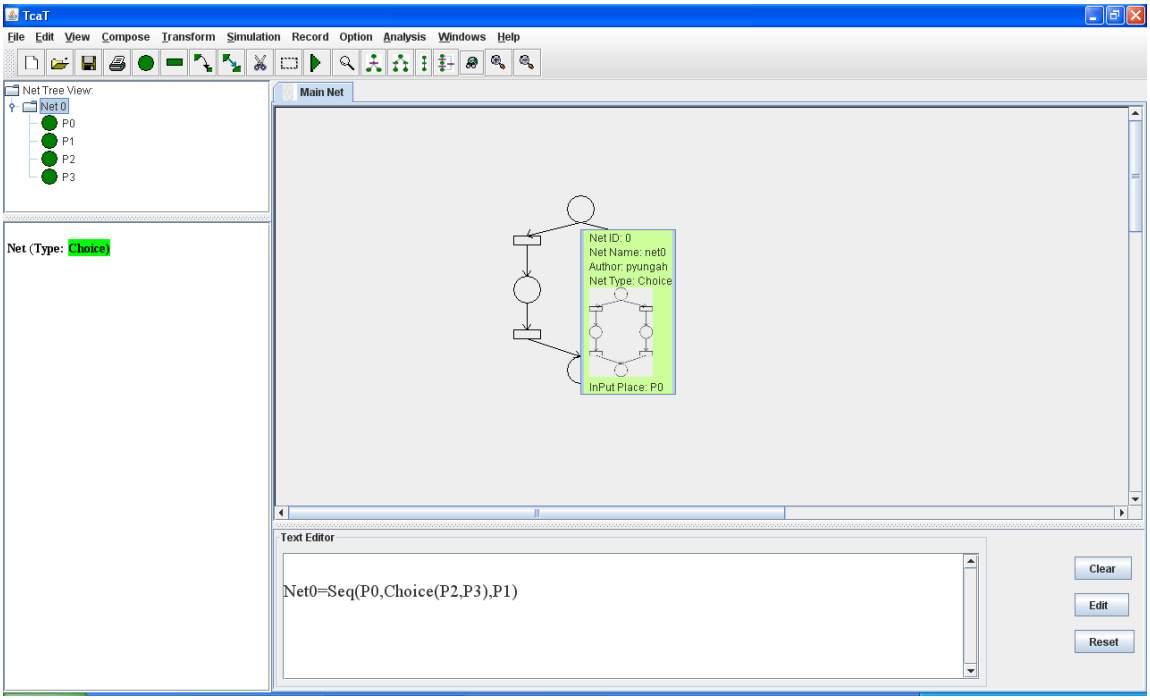


Figure 23: Net0 and the Tool Tip Displaying its Information

An author can modify attributes of template through a popup menu invoked by clicking the right mouse button after selecting the net.

Place ID and name, transition ID and name, and arc ID and name are also automatically assigned in the same way as net ID and name are assigned. We can setup, modify and see the attributes of each component of net; place, transition and arc. TcAT maps a place with the content of a document, as shown in Figure 24. The media type of content can be text, image, audio, video, web page, and several other kinds of files. Figures 24 (a) to (e) show that the image file is mapping to a place. The content is displayed in the content panel and tool-tip. Figure 24 (f) presents the audio file is mapping to a place and Figure 24 (g) shows the video file is mapping to a place.

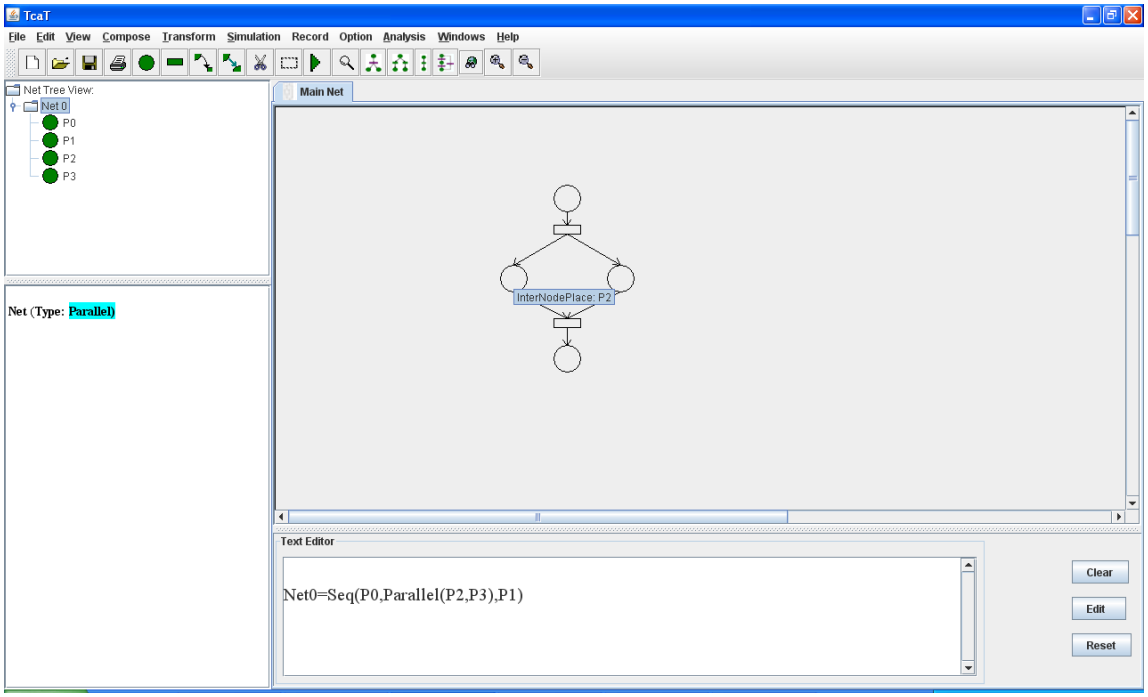


Figure 24 (a): Place p2 with Information

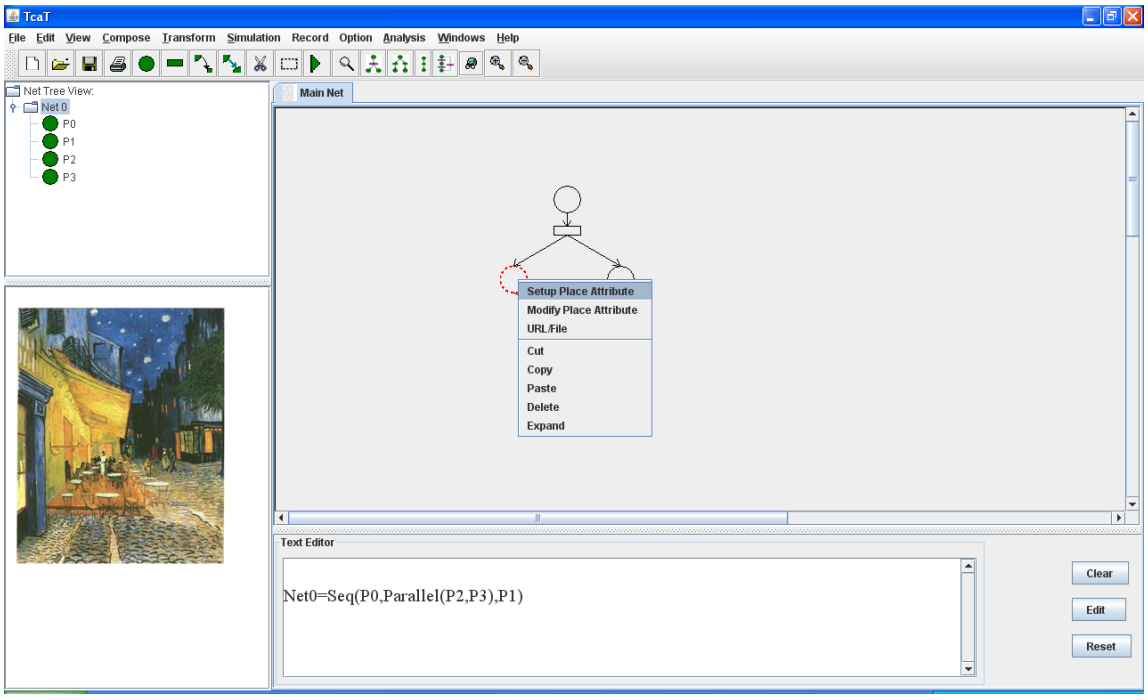


Figure 24 (b) Place p2 with Popup Menu

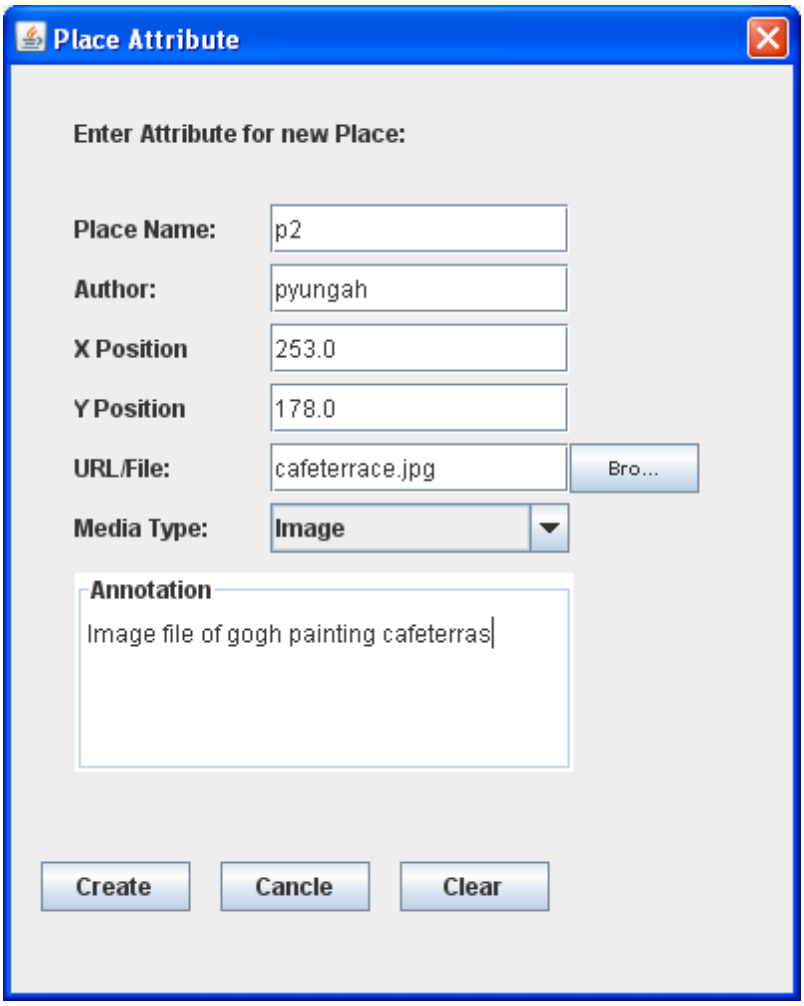
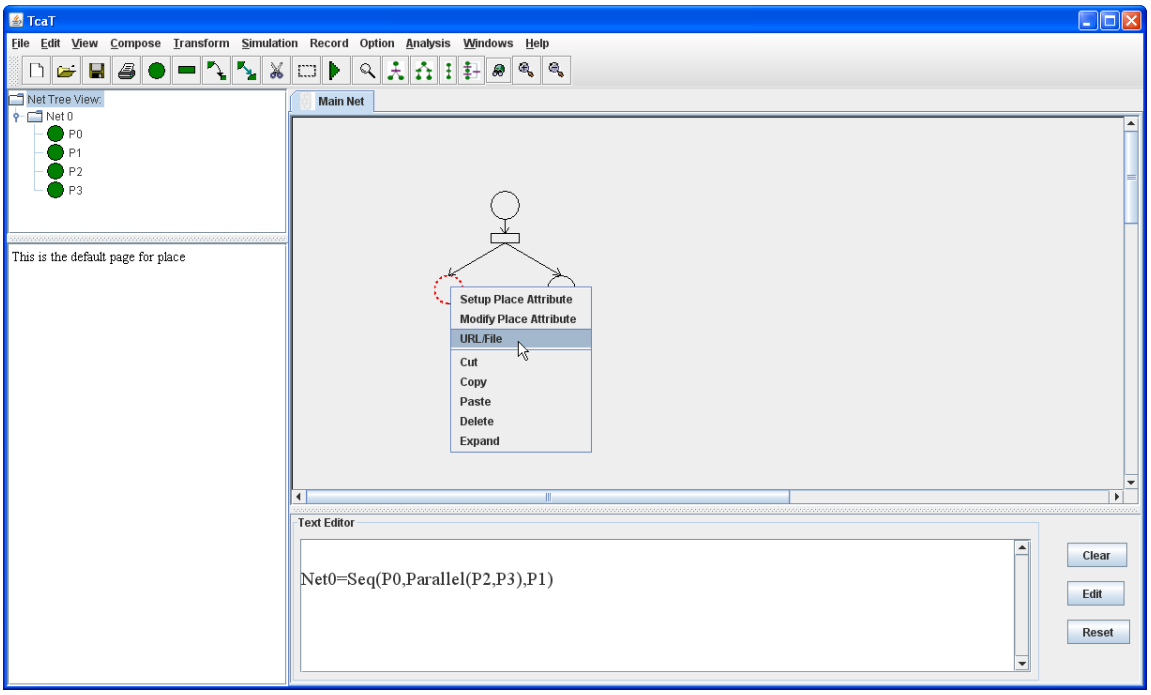


Figure 24 (c): Set up Place p2 Attributes



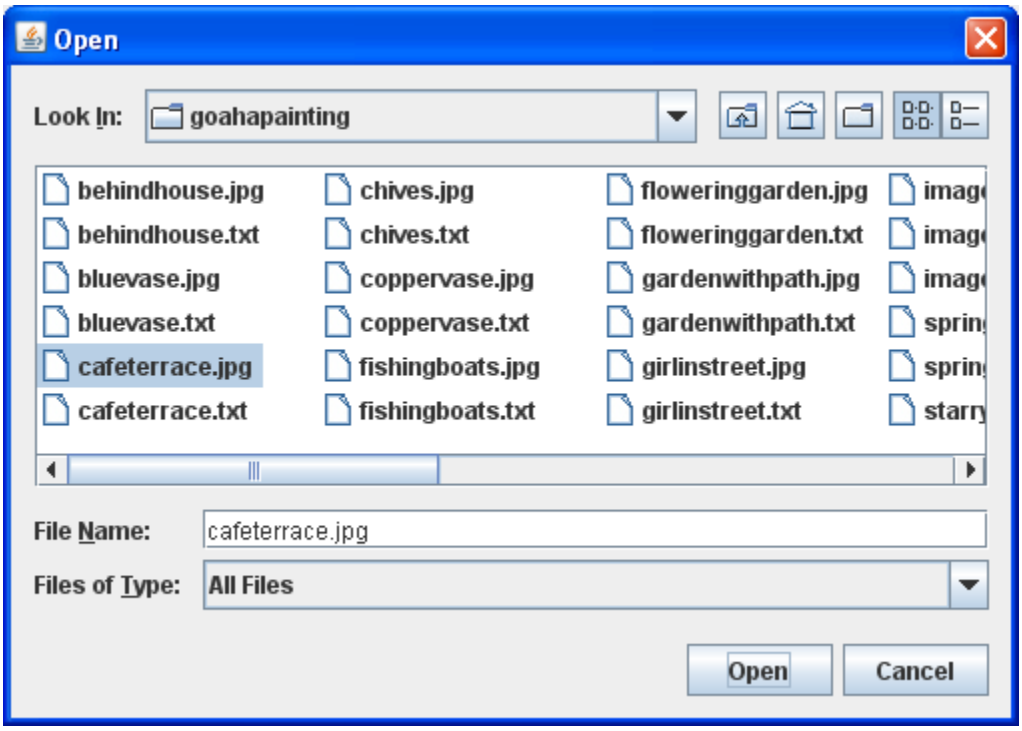


Figure 24 (d): Map a Content File with a Place p2

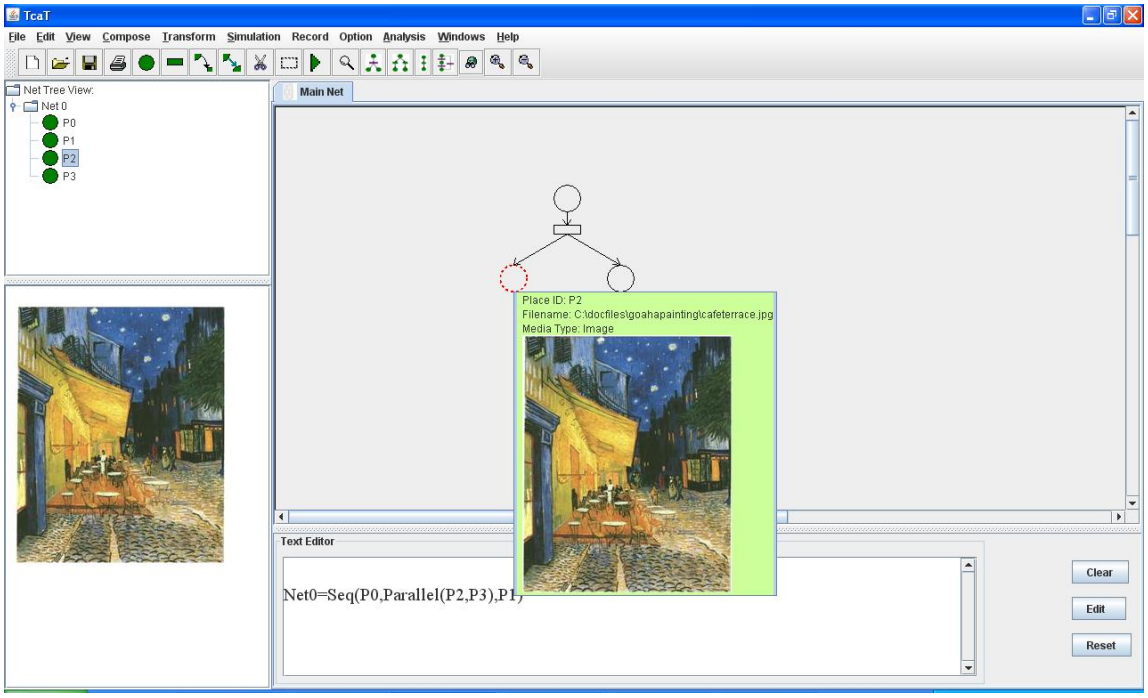


Figure 24 (e): Content Mapping

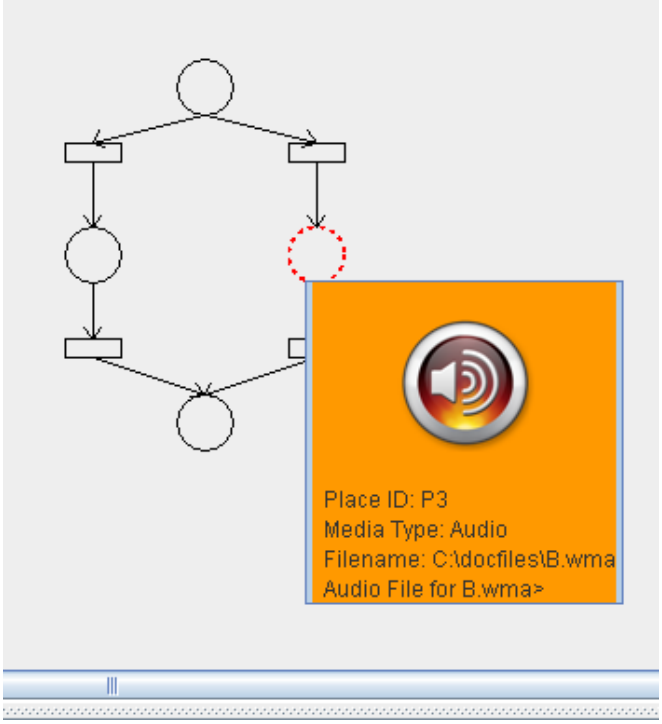


Figure 24 (f): The Mapping Place with Audio File and its Tooltip

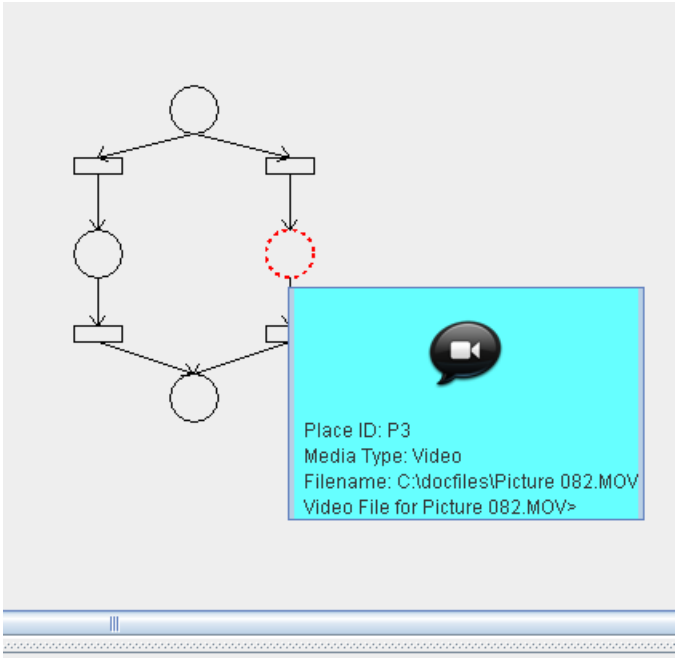


Figure 24 (g): The Mapping Place with Video File and its Tooltip

TcAT includes a document editor for easy editing of content associated with places. For editing complex document types that TcAT does not handle natively, double clicking the selected place invokes other applications such as office programs, web browsers, image viewer and media players. According to type of content, the relevant application is invoked. For example, if the content is the “ppt” (power point) file type, TcAT invokes the power point application as presented at Figure 25 (a). Figure 25 (b) shows that TcAT presents the relevant web page in a web browser if the content contains a URL. As a default, a text editor is invoked.

Figure 26 presents an author making the text content for a selected place (i.e., “place 2”, red dot circle) by invoking the text editor. Figure 27 shows the content for “place 2” in the content layout panel and a tool-tip.

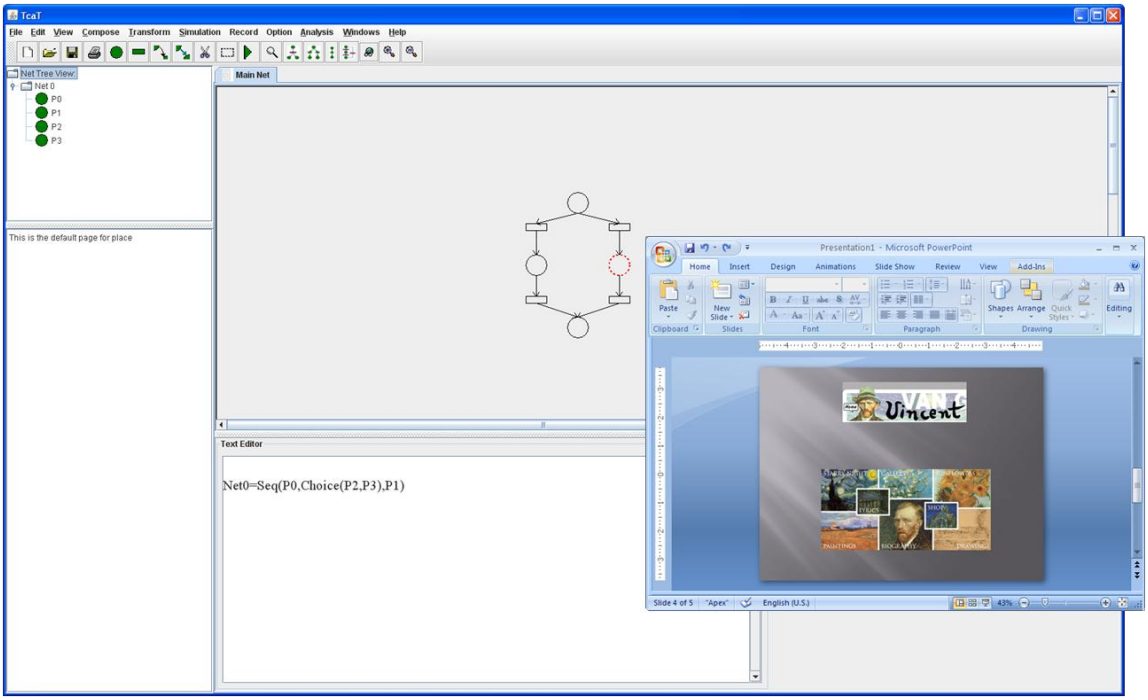


Figure 25 (a): Invoking the Power Point Application

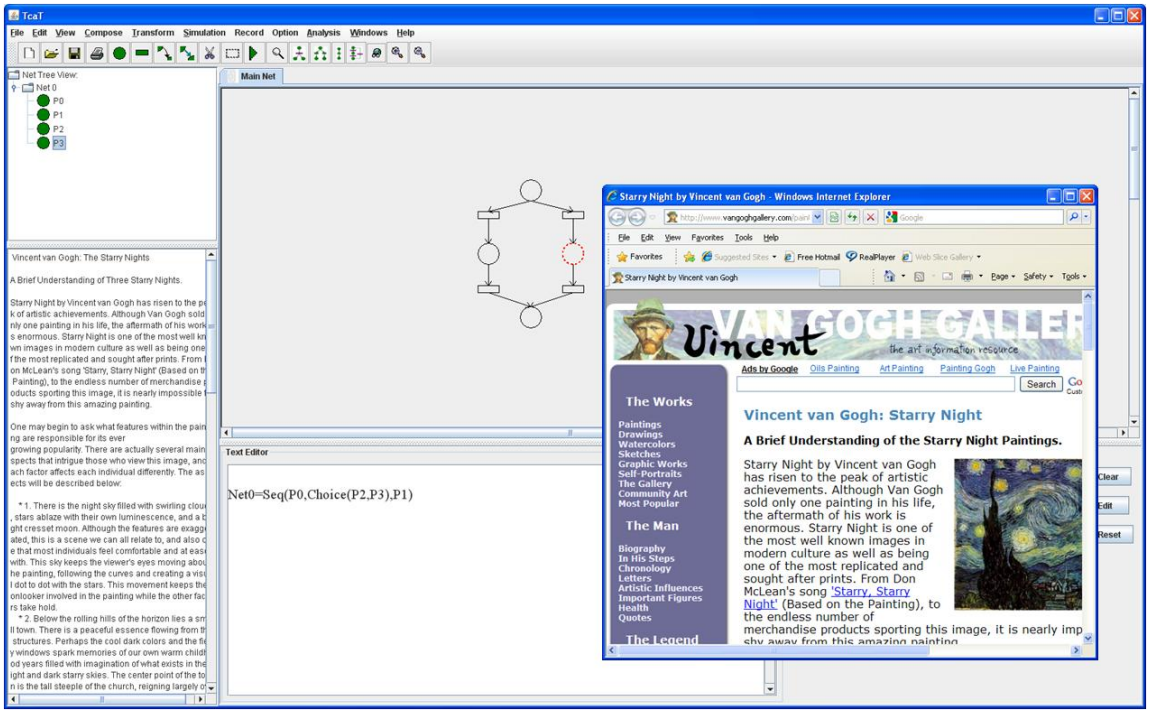


Figure 25 (b): Invoking the Relevant Web Page in a Web Browser

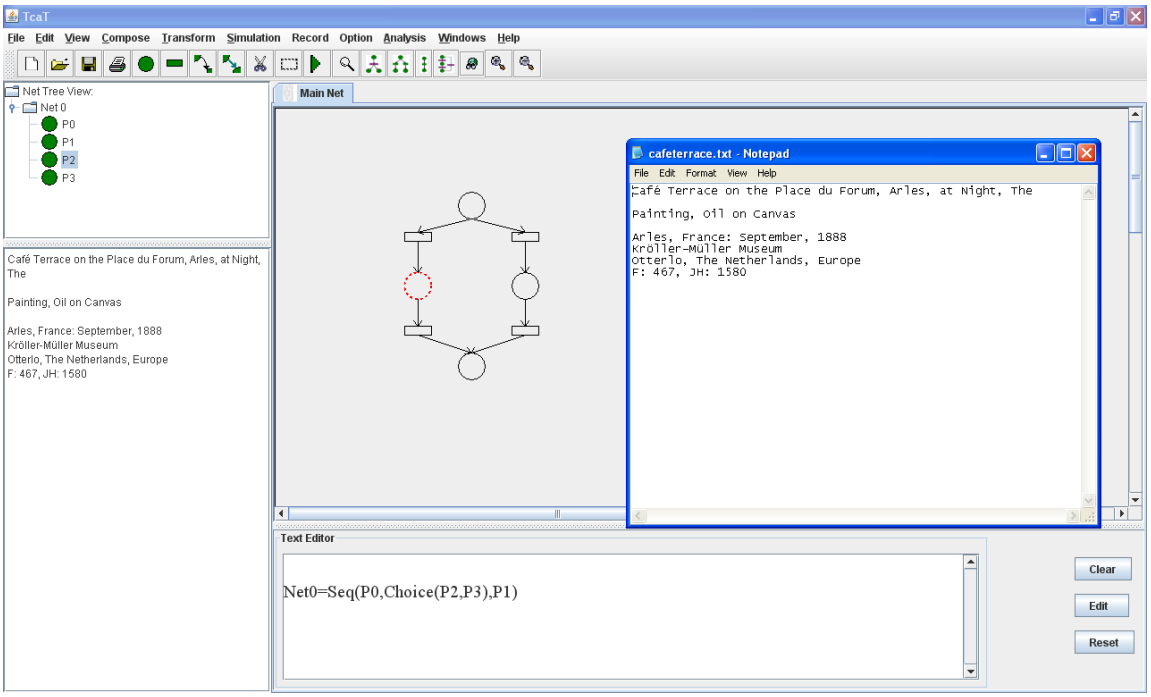


Figure 26: Making Content for a Place with the Default Text Editor

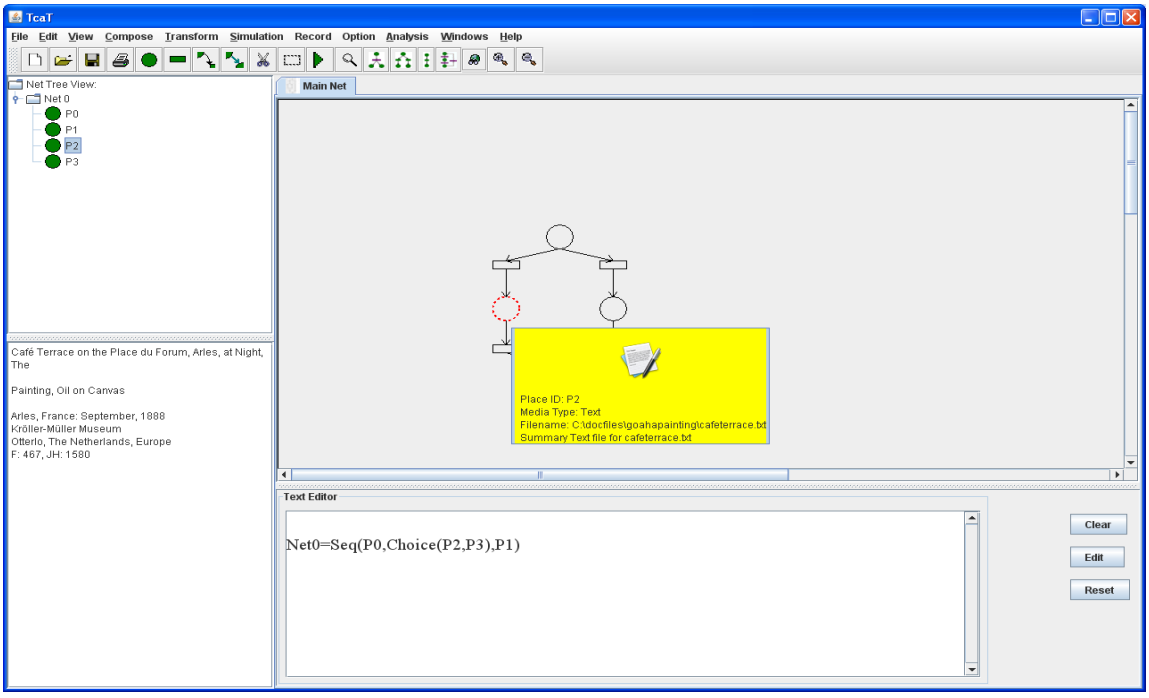


Figure 27 Content View for a Place with Tool Tip and Content Panel

3.3.4 Net Composition

We compose nets with four composition operations; sequence, choice, parallel and repeat through graphical interactions. After selecting the nets which we want to compose, we click “compose” item in popup menu invoked by clicking the right mouse button and then chose one composition operation. After we providing the information that TcAT requires to compose, TcAT automatically composes the selected nets. Figure 28 presents the composition process.

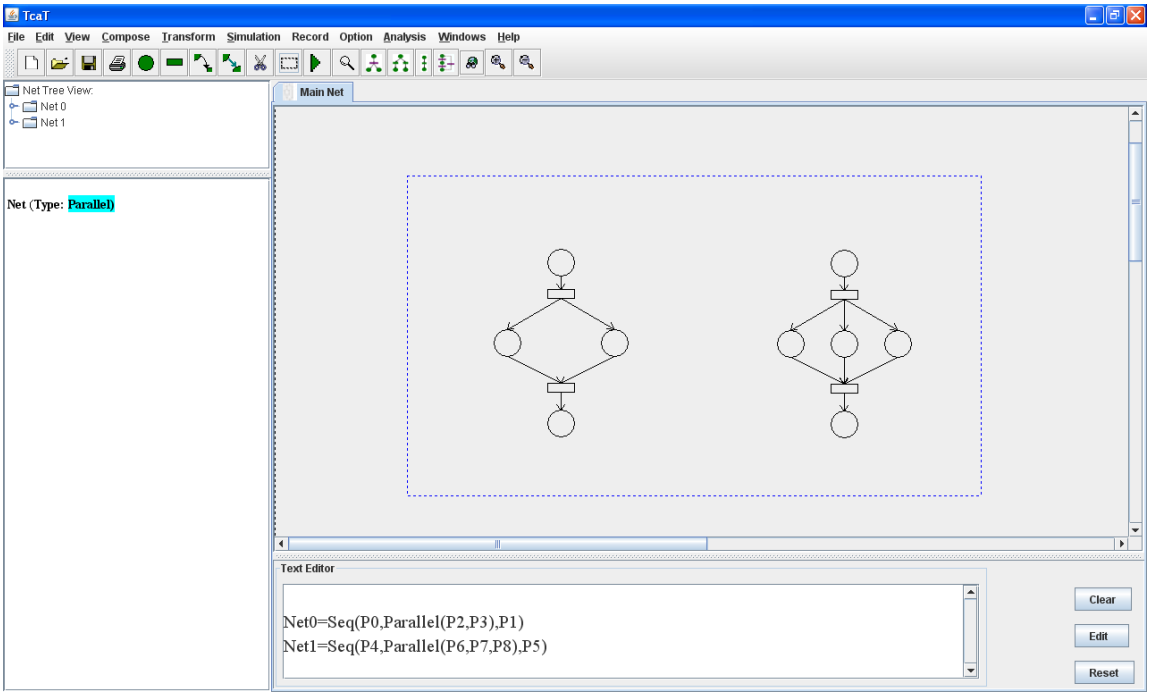


Figure 28 (a): Selecting Subnets to be Composed

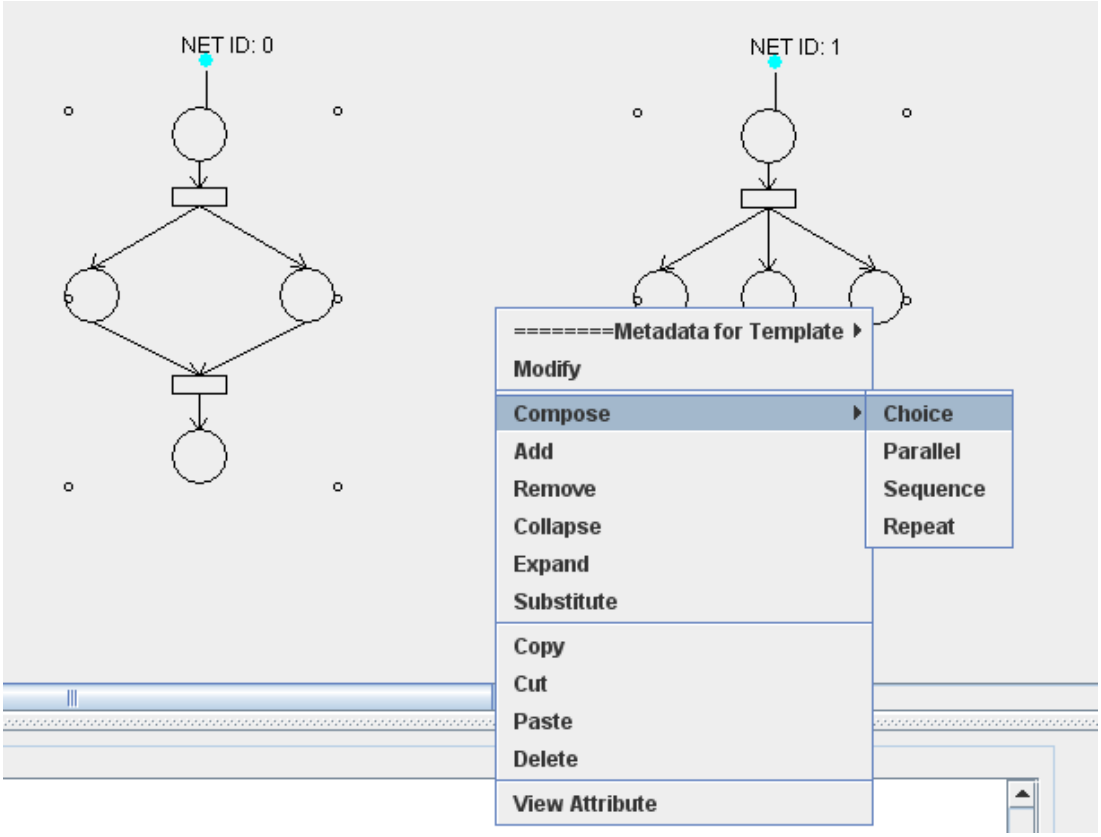


Figure 28 (b): Choosing an Operation at Popup Menu

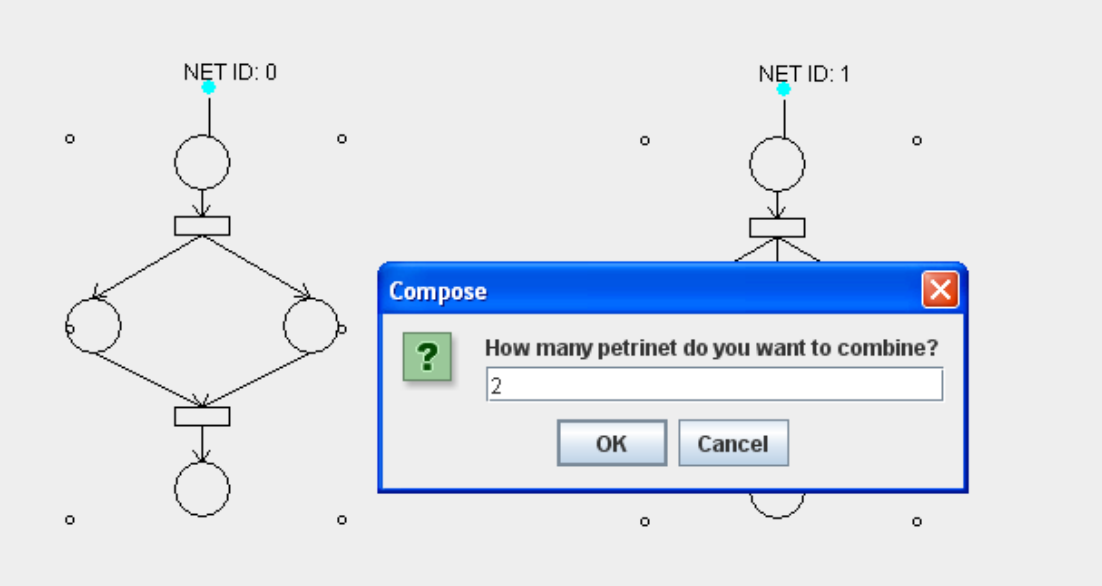


Figure 28 (c): Providing Number of Subnets to be Combined

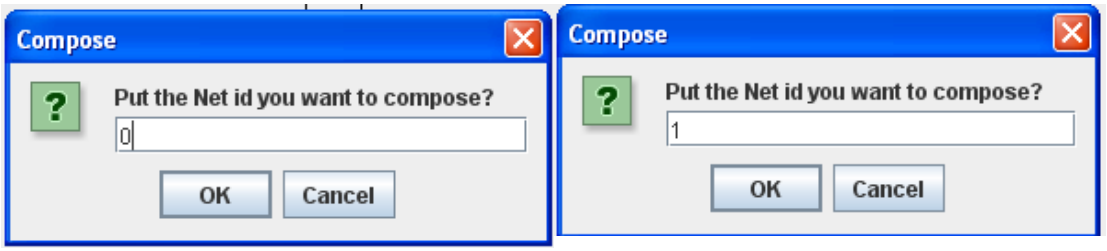


Figure 28 (d): Providing Net ID for Each Subnet

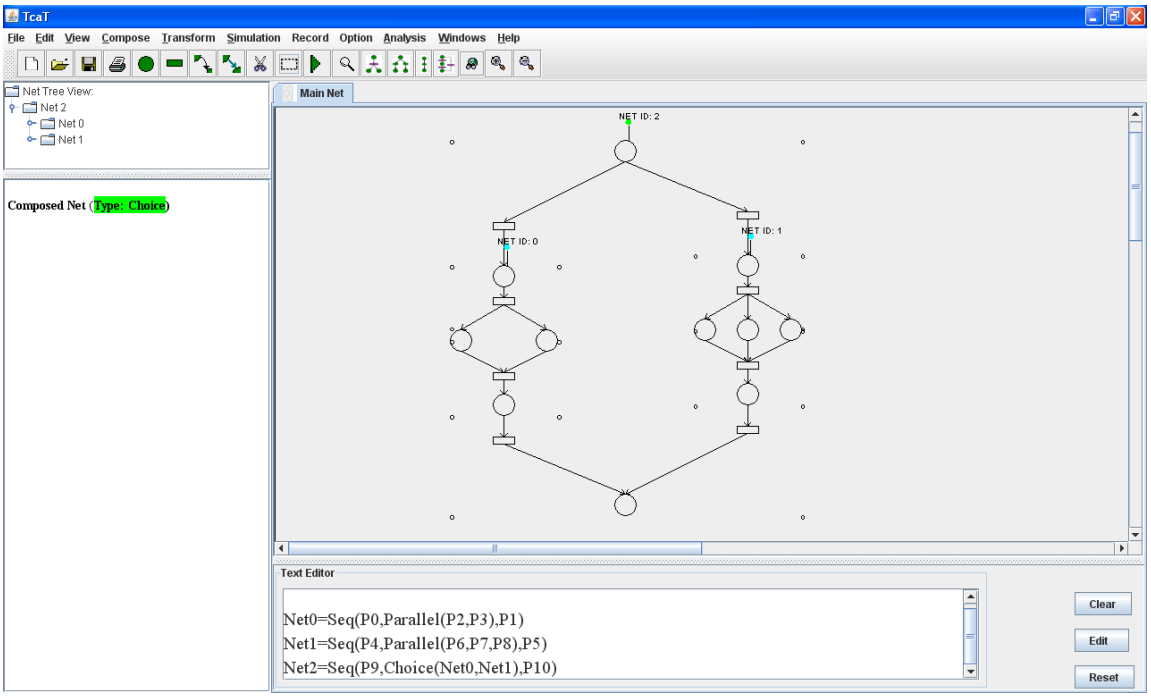


Figure 28 (e): Result of Composition Process

3.3.5 Modifying Net

After creating a net, we can modify the net by adding places to the net and removing places from the net. When adding places to the selected net, we enter the number of place want to add. This addition occurs automatically. Figure 29 shows the addition of one place to net0.

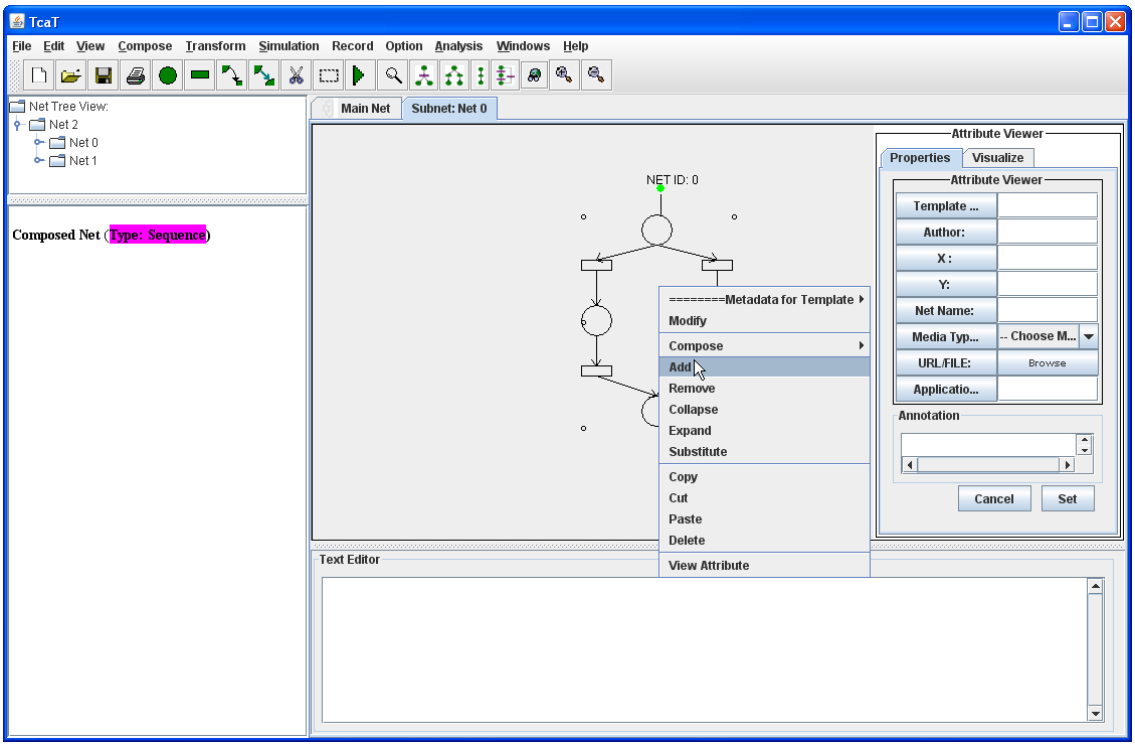


Figure 29 (a): Choosing Add Function at Popup Menu

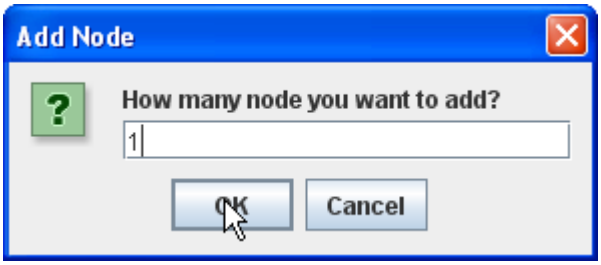


Figure 29 (b): Providing Number of Place to be Added

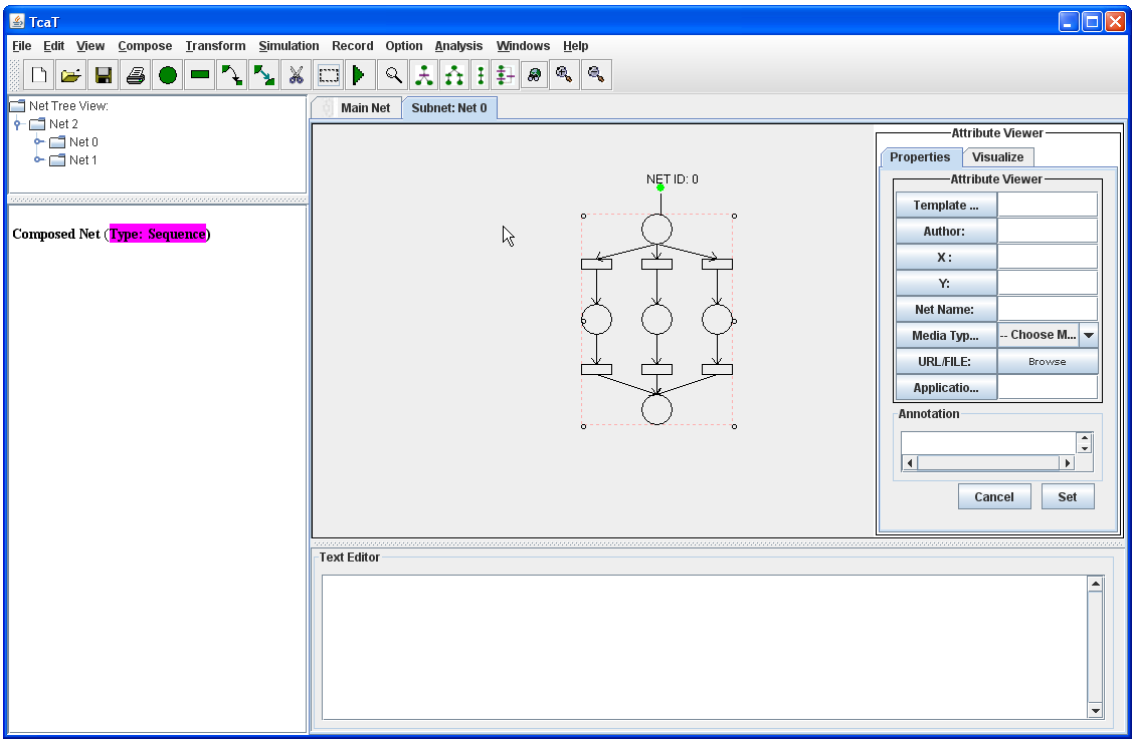


Figure 29 (c): Result of Adding a Place at Net

3.3.6 Tabbed Panel

To reduce display complexity of a large net, a tabbed panel displays the main net, as well as its subnets. A net is collapsed as a special form (i.e., representative form) of place and the original net is automatically drawn at the next tab panel as shown in Figure 30(b). As presented in Figure 30(c), authors can modify the net at the next tab panel. When the collapsed net is expanded, the modified net is drawn at the original panel as shown in Figure 30(d). This feature simplifies net editing and management, as well as reducing display complexity of large and complex nets. For nets with multiple instances, the modification affects the every instance of the net.

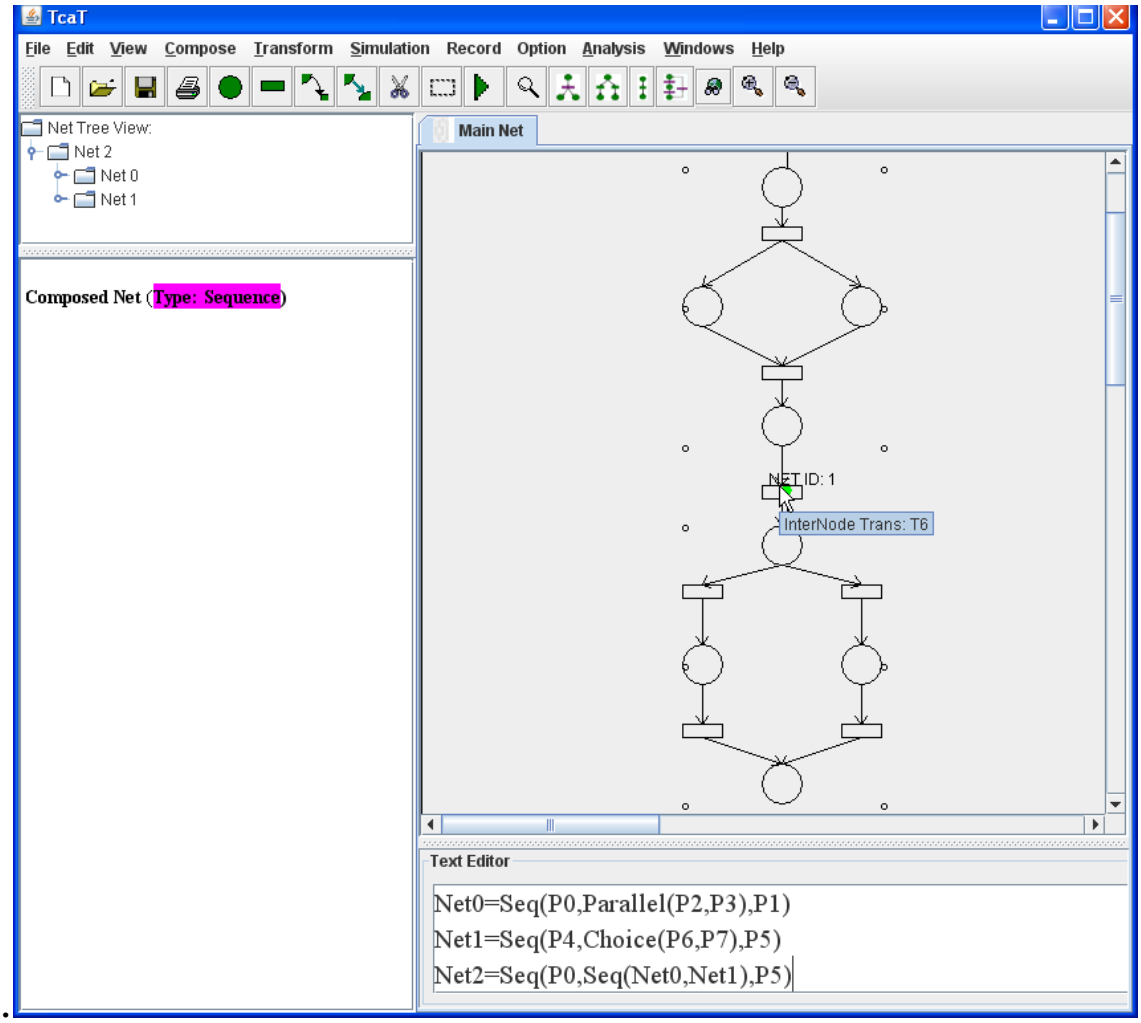


Figure 30 (a): Original View at Main Net

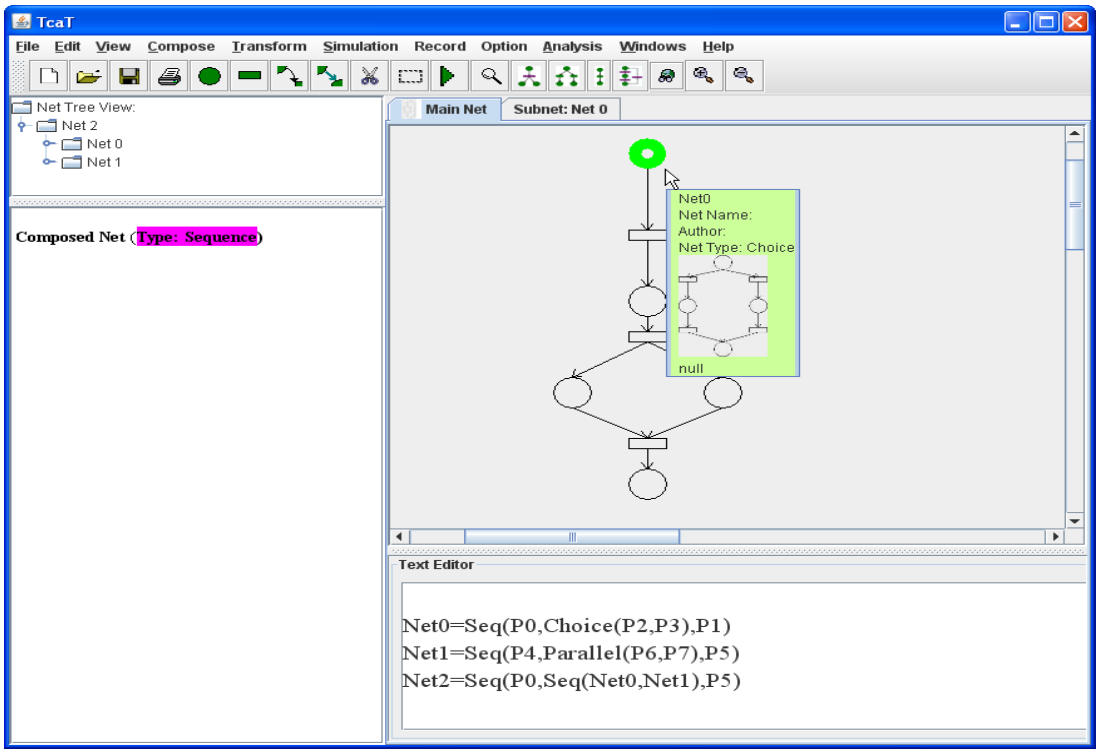


Figure 30 (b): Collapsed Net at Main net

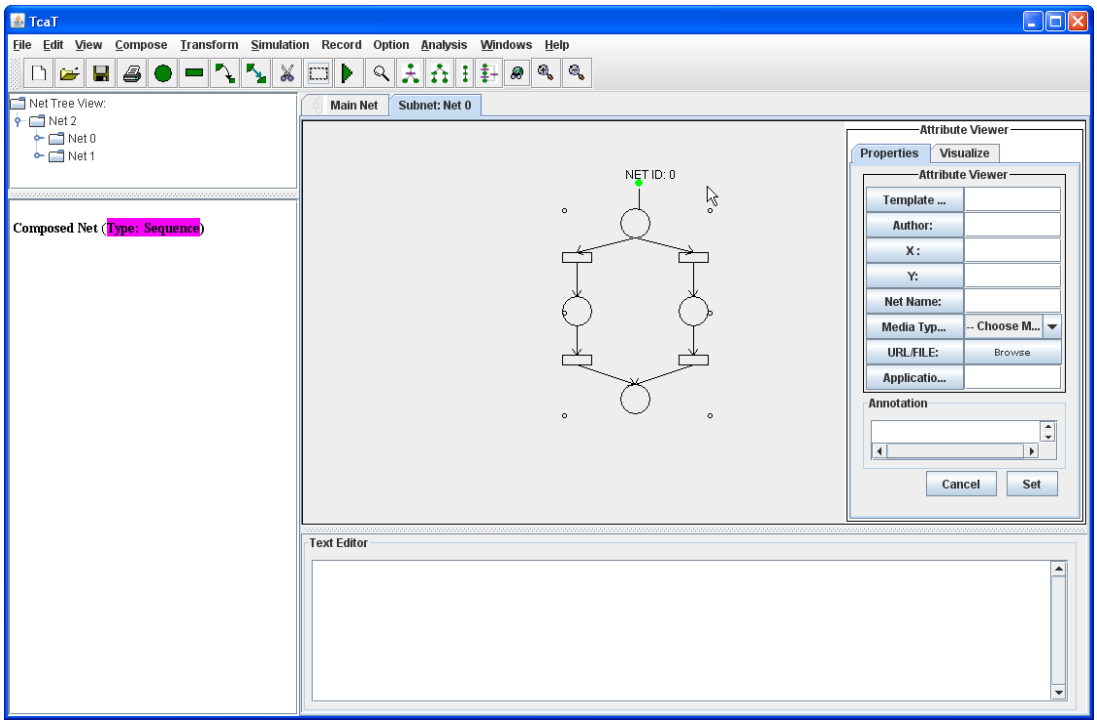


Figure 30 (c): Subnet at Next Tab Panel

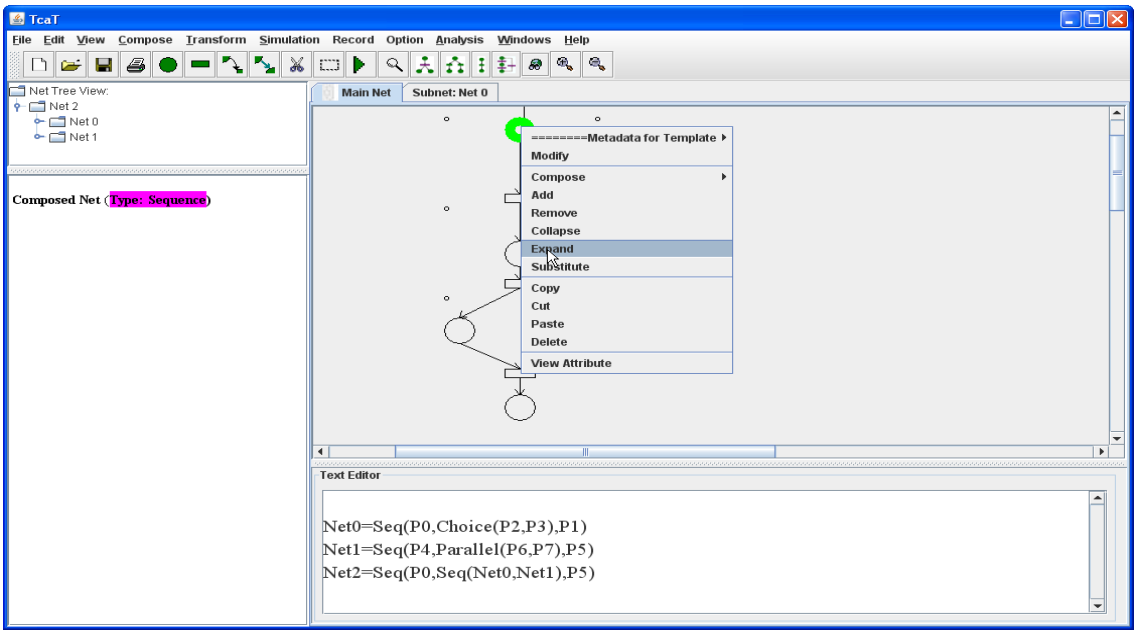


Figure 30 (d): Expanding a Representative Place

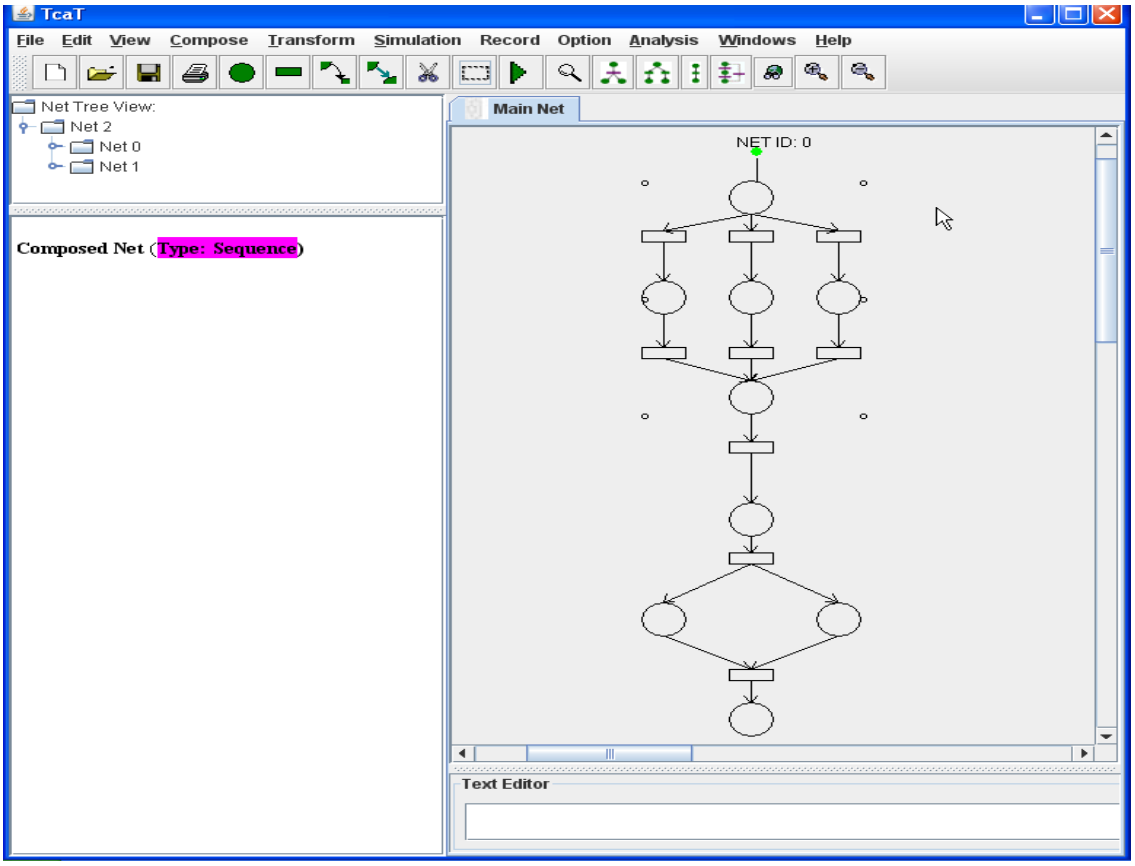


Figure 30 (e): Expanded View at the Main Net after Adding a Place

3.3.7 Textual Editor for Authoring Language

TcAT's text editor doubles as a textual authoring environment for caT component nets. TcAT converts the textual specification to the graphical form automatically and vice versa. Figure 31 shows an example of automatic conversion between the textual language and its graphical equivalent using provided templates. In this example, the author has combined the two nets shown in Figure 31(a) into a sequence displayed in Figure 31(b).

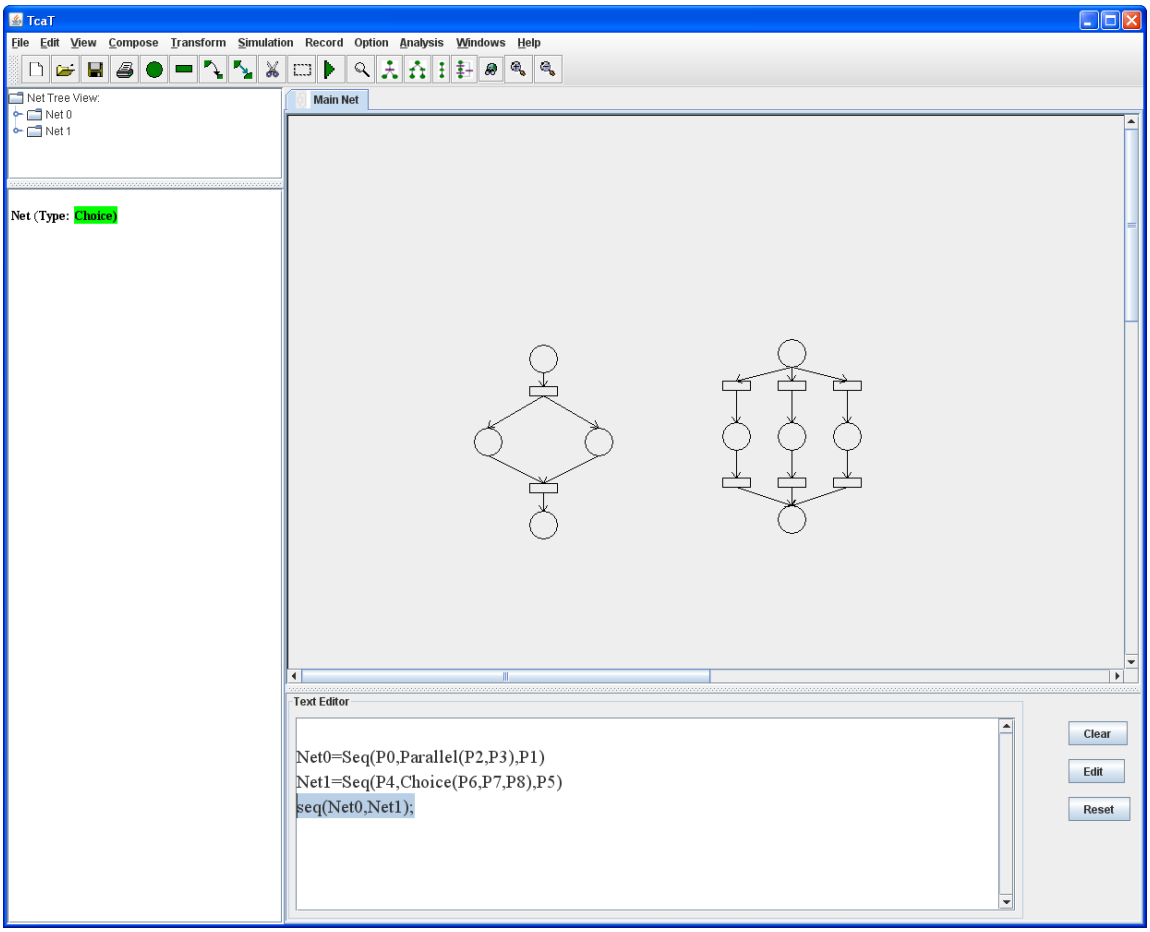


Figure 31 (a): Combining Two Nets by Using Textual Language

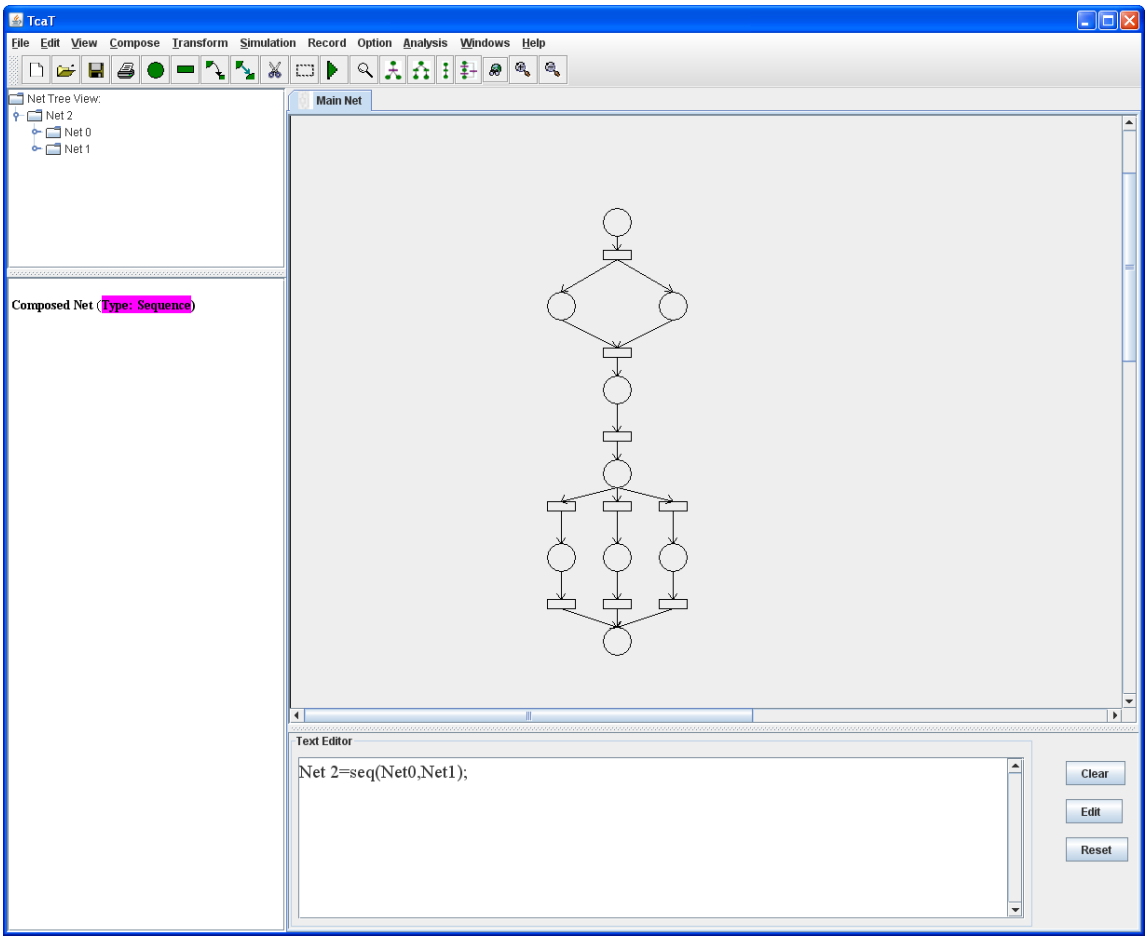


Figure 31 (b): A Combined Net Specified with the Textual Language

3.3.8 Search

The provided templates can be further customized by authors to suit their specific needs. Authors may also search for specific templates. TcAT identifies the location of these templates within a large net as shown in Figure 32.

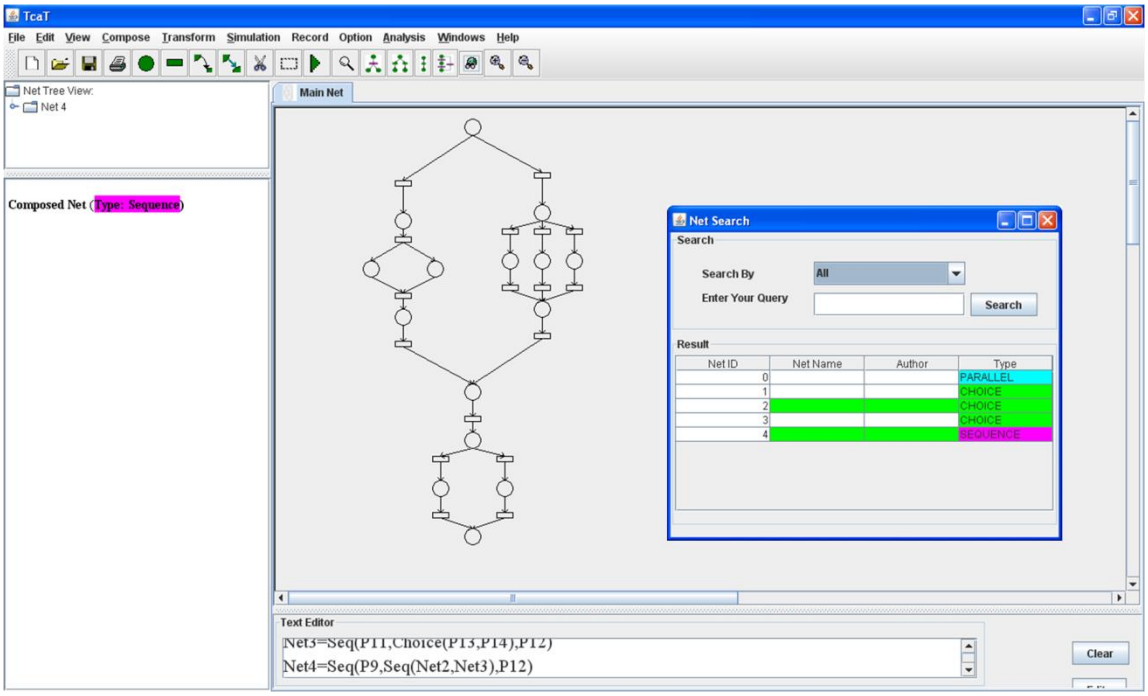


Figure 32 (a): Searching Nets via Search Window

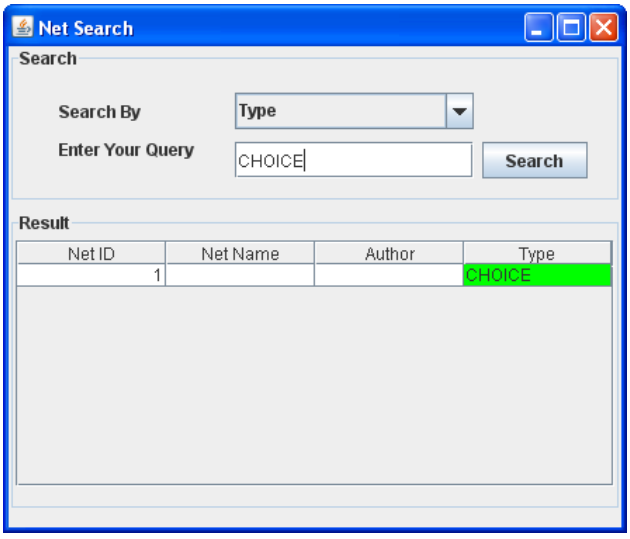


Figure 32 (b): Search Window

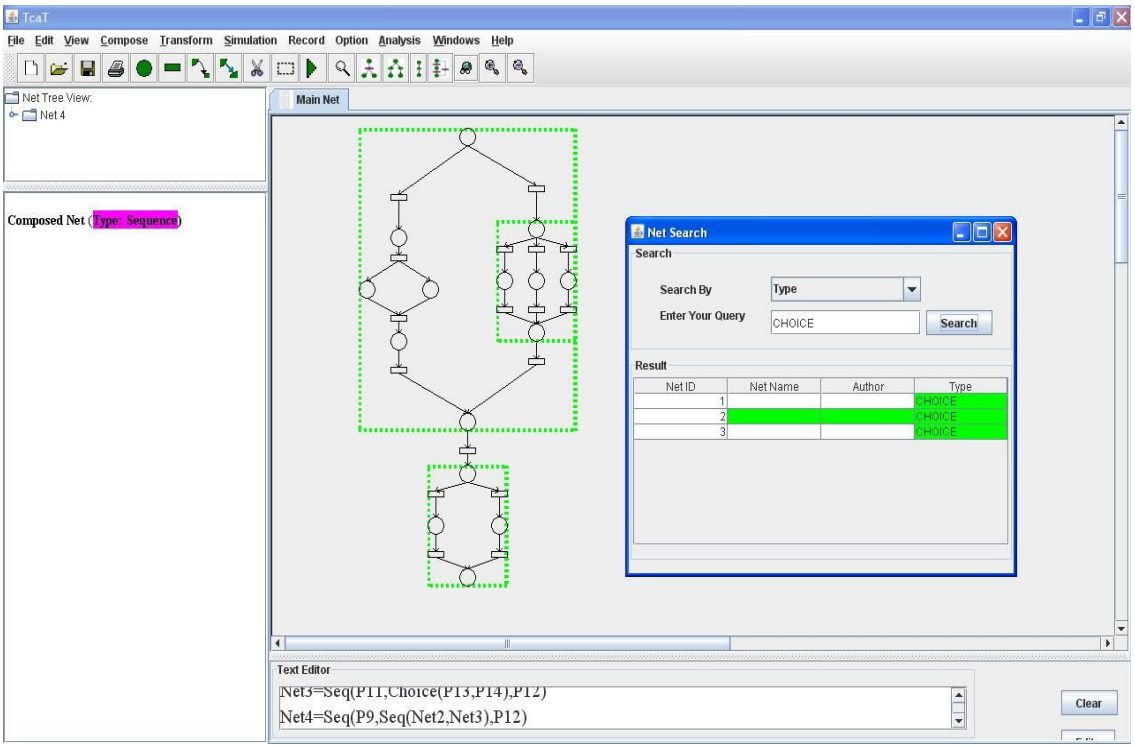


Figure 32 (c): Searching Nets by Net Type

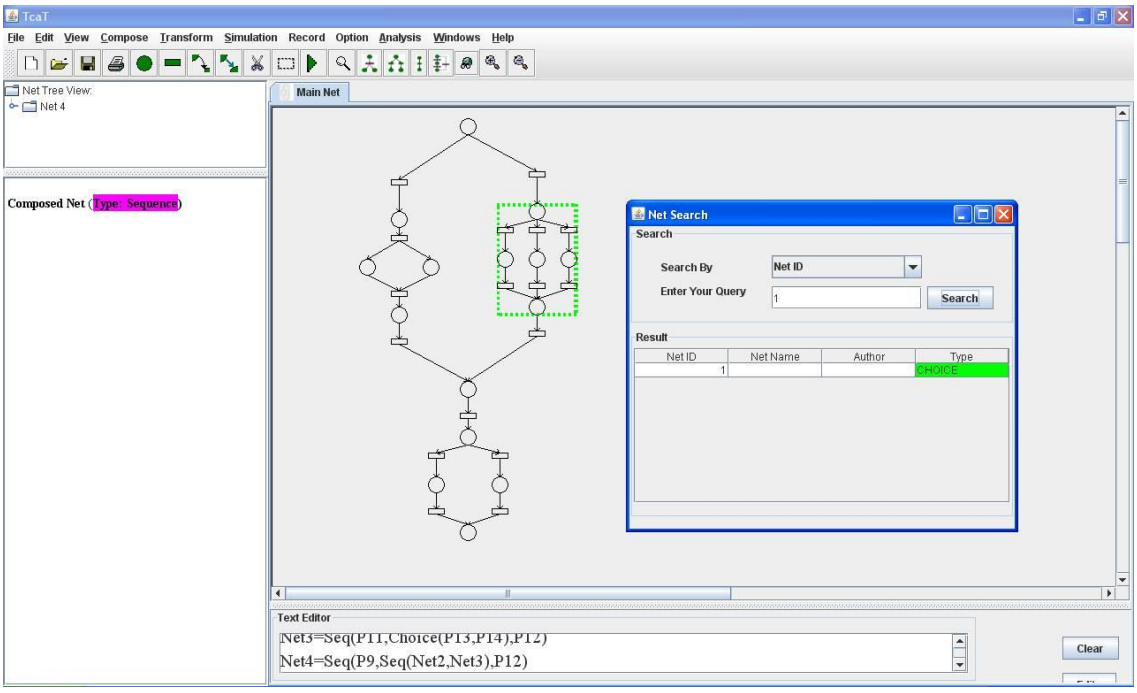


Figure 32 (d): Searching a Net by NetID

CHAPTER IV

SYSTEM AND IMPLEMENTATION DETAIL

This chapter provides system architecture of TcAT system, specific description of TcAT's interaction, and implementation details for TcAT.

4.1 System Architecture

TcAT consists of three components: Petri net engine, database component, and display component. Figure 33 shows the system architecture of TcAT.

As presented in Figure 34, the Petri net engine adds a component net mechanism and net transformations to the high-level net component.

Each generated net, including template and its information, are stored in net specification repositories. Each place, transition, and arc stored in place, transition and arc repositories. The database manager and library finder are used to store and search for components in the net library. TcAT searches for nets/templates in the net specification repository to compose a net. After generating the net, the Petri net engine sends this net to the authoring tool for displaying the graphical view for human consumption.

The display component consists of a graphical editor, textual editor, analyzer, parser, and automatic layout computation component as shown Figure in 35. The textual authoring language is automatically converted to a Petri net using a textual language parser and an automatic layout component calculator for positioning of net elements. Figure 36 shows this process.

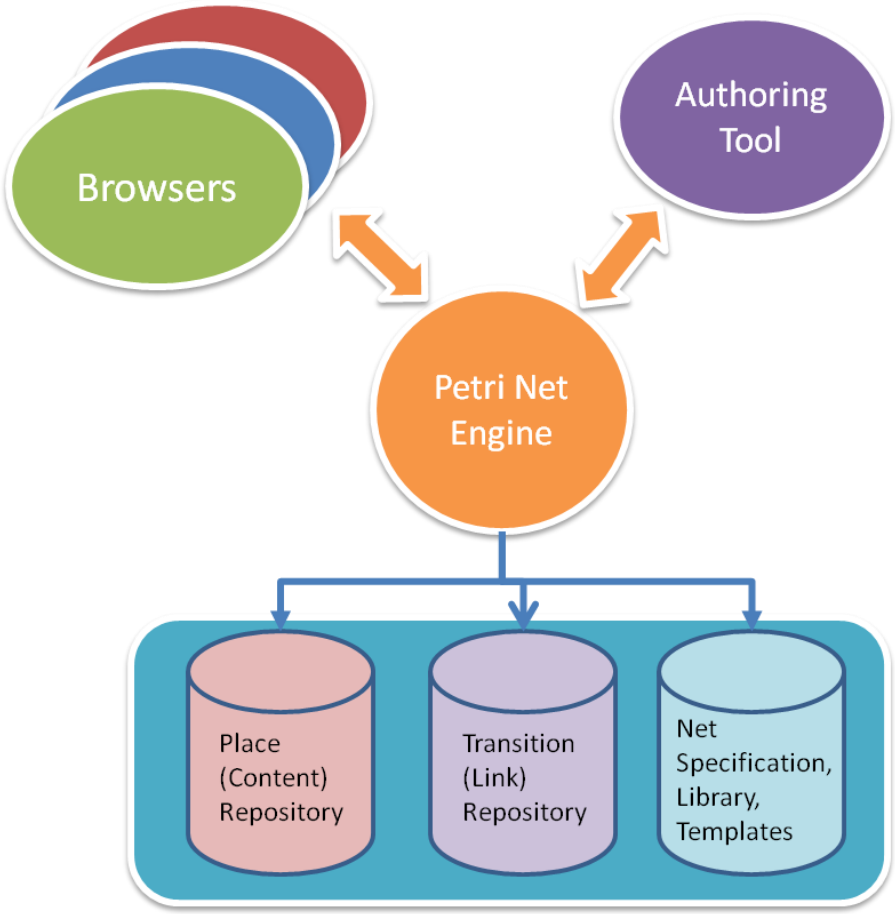


Figure 33: System Architecture

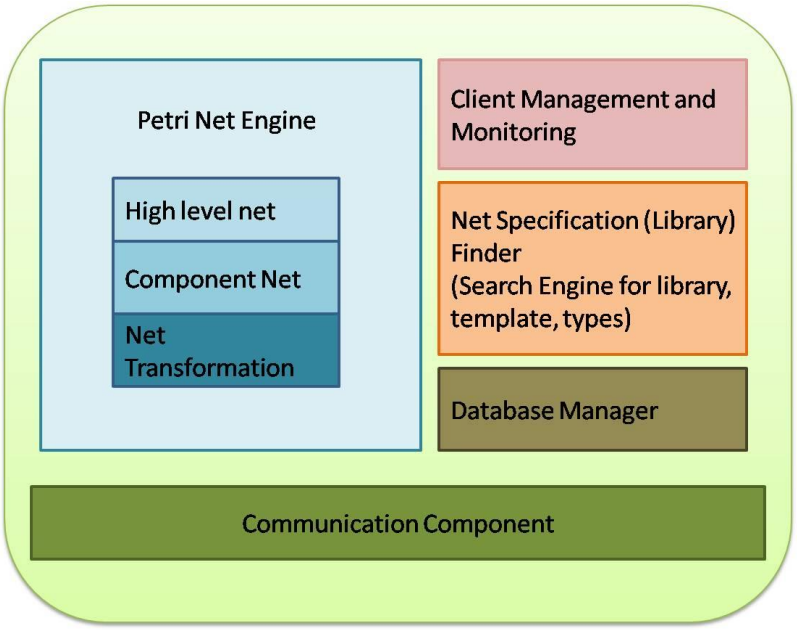


Figure 34: Components of Petri Net Engine

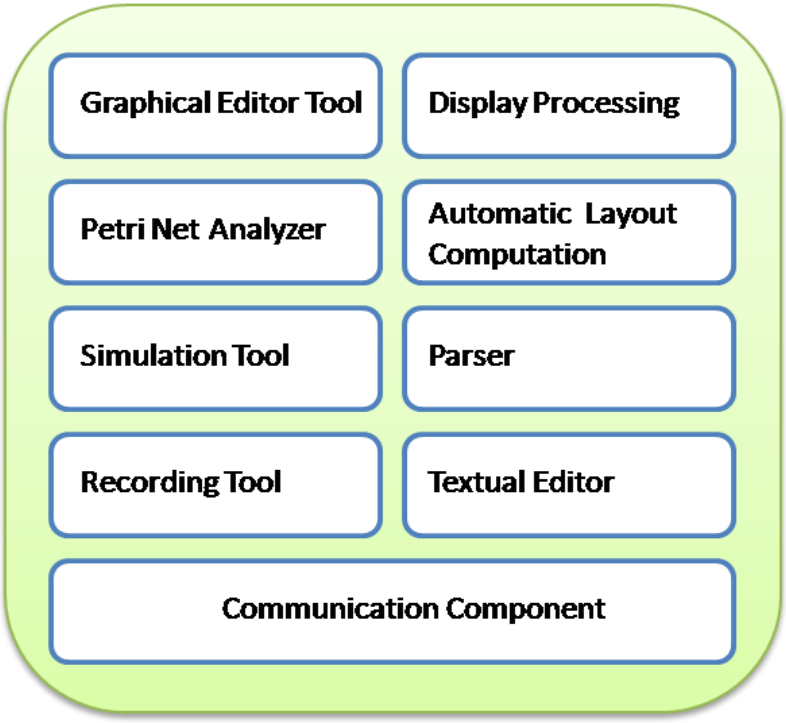


Figure 35: Components of Authoring Tool

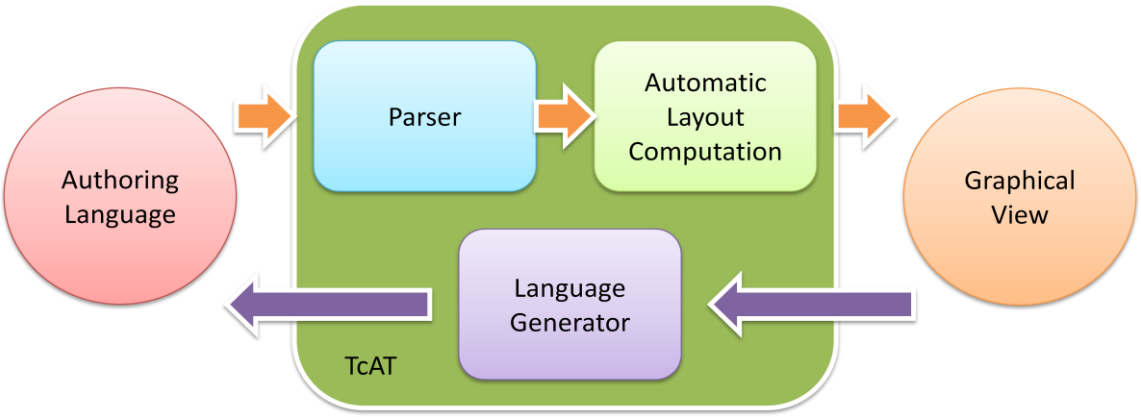


Figure 36: Processing for Authoring Language to Graphical View

4.2 System Details

4.2.1 Interaction with Main Menu and Image Icons

As shown in Figure 37, the main menu bar, consisting of 11 menu items, is located at top of a main window. By default view, the toolbar that groups a 19 buttons with image icons is underneath the menu bar. By dragging the tool bar to another edge of a window or outside the window, the author can change the location of the tool bar.

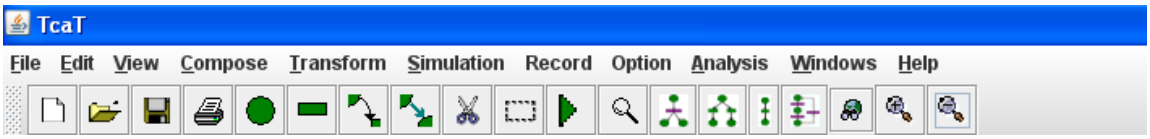


Figure 37: Items of TcAT's Main Menu

Figure 38 shows the file menu and its sub menu items whose brief description is at Table 2. As shown in Figure 38, for keyboard operation, TcAT's main menu supports mnemonics (i.e., F for File) to navigate the menu hierarchy and accelerators for shortcuts (i.e., Alt-1 for open).

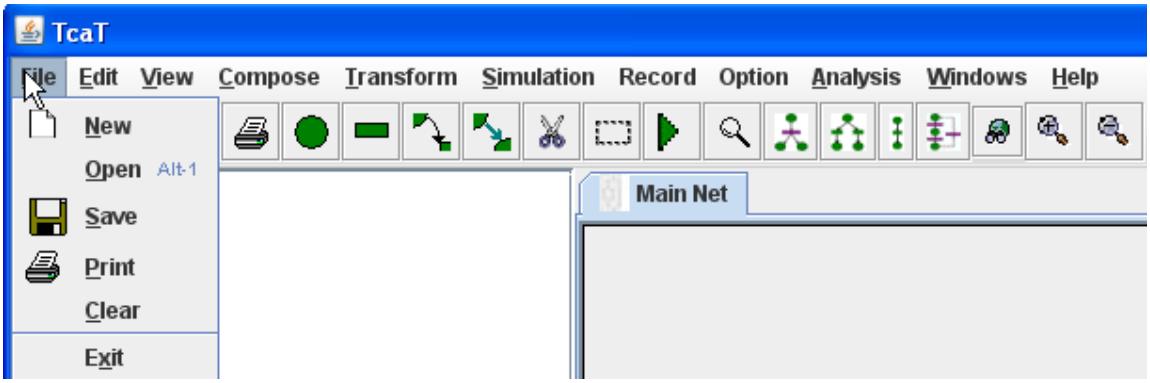


Figure 38: File Menu and Its Sub Menu of TcAT

Table 2: Sub Menu Items of File Menu

Main Menu	Sub Menu	Description
File	New	Make a new hypertext document
	Open	Load the selected document
	Save	Save the current running document
	Print	Print the current running document
	Clear	Remove the current running document from canvas
	Exit	Exit the authoring tool

The edit menu and its sub menu items are presented in Figure 39. Table 3 provides a brief description of each item.

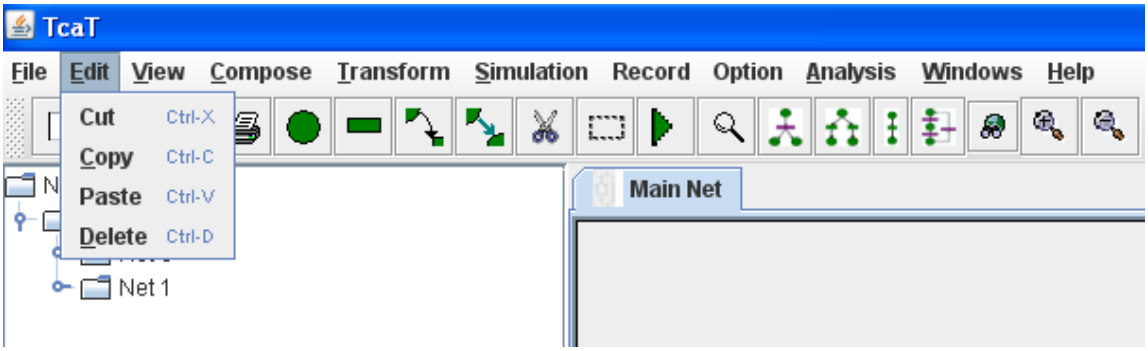


Figure 39: Edit Menu Item and Its Sub Menu of TcAT

Table 3: Sub Menu Items of Edit Menu of TcAT

Main Menu	Sub Menu	Description
Edit	Cut	Cut selected objects
	Copy	Copy selected objects
	Paste	Paste copied objects
	Delete	Delete selected objects

The view menu provides options for display of the template list panel as presented in Figure 40 and table 4. The author chose the “Template Lists” item to view the template list panel. By default, TcAT does not display the current template list panel.

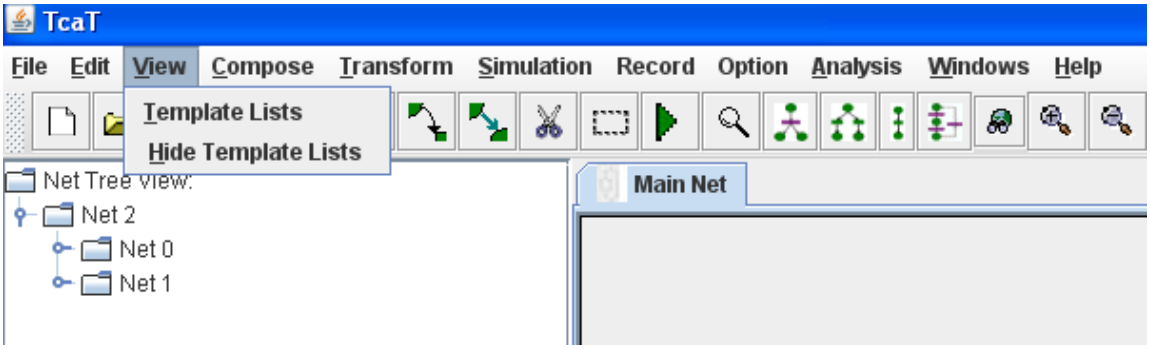


Figure 40: View Menu and Its Sub Menu of TcAT

Table 4: Sub Items of View Menu of TcAT

Main Menu	Sub Menu	Description
View	Template List	Display the current template list panel
	Hide Template List	Hide the current template list panel

The compose menu and its sub menu items are presented in Figure 41 and table 5. Figure 42 and table 6 presents the transform menu and its sub menu items.

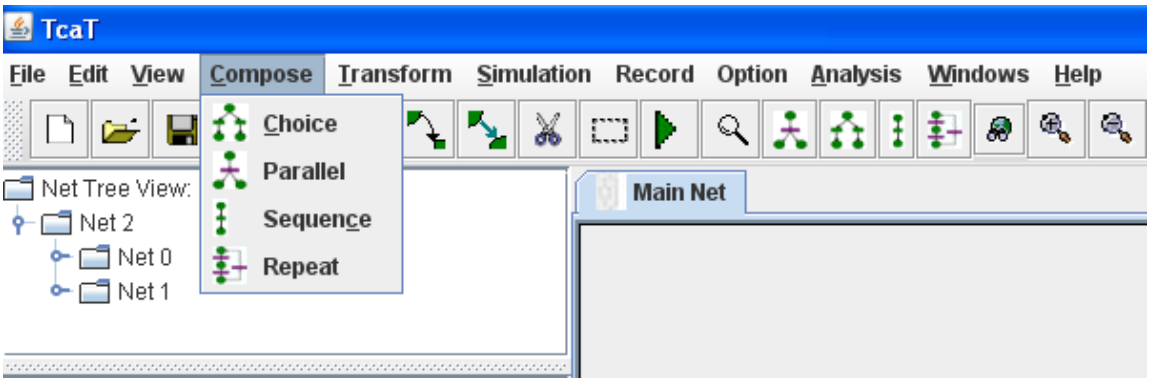


Figure 41: Compose Menu and Its Sub Menu of TcAT

Table 5: Sub Items of Compose Menu of TcAT

Main Menu	Sub Menu	Description
Compose	Choice	Compose net by using choice composition
	Parallel	Compose net by using parallel composition
	Sequence	Compose net by using sequence composition
	Repeat	Compose net by using repeat composition

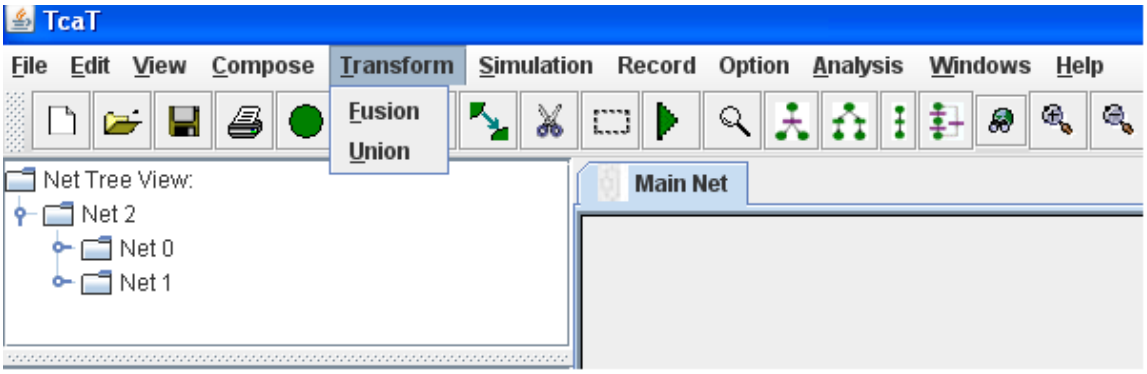


Figure 42: Transform Menu and Sub Menu of TcAT

Table 6: Transform Menu and Its Sub Menu of TcAT

Main Menu	Sub Menu	Description
Transform	Fusion	Make a fusion operation for net transformation
	Union	Make a union operation for net transformation

Table 7 presents names and the brief functional descriptions for image icons in the tool bar.

Table 7: Image Icons and Its Description

Icon	Name	Description
	Place	Create a Place at the position the mouse is clicked
	Transition	Create a Transition at the position the mouse is clicked
	Arc	Create an arc
	Move	Move objects/group of objects/net
	Cut	Delete object/group of objects/net
	Select	Select Nets and Net elements to manipulate
	Play	Play a simulation
	Parallel Template	Create a Petri net whose type is parallel
	Choice Template	Create a Petri net whose type is choice
	Sequence Template	Create a Petri net whose type is sequence
	Repeat Template	Create a Petri net whose type is repeat
	Search	Search a net by using meta data
	Zoom In	Zoom in to get a larger view
	Zoom Out	Zoom out to get a smaller view

4.2.2 Popup Menus

Along with the main menu bar, TcAT provides several popup menus which are brought up by pressing or clicking the right mouse button. Figure 43 shows the popup menu after a selecting a net. This popup menu is to manipulate selected nets.

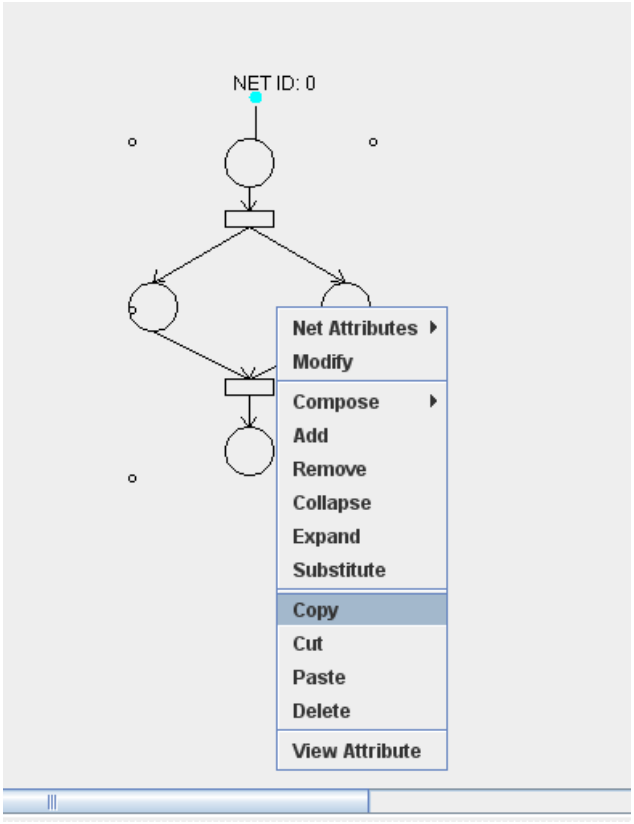


Figure 43: Popup Menu after Selecting a Net

For manipulating for places, transitions, and arcs, TcAT brings up each popup menu. Figure 44 presents the popup menu after selecting a place.

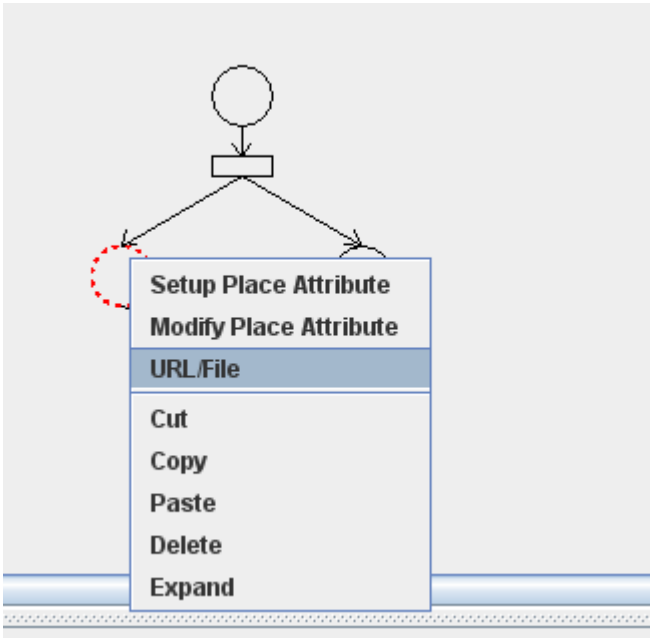


Figure 44: Popup Menu after Selecting a Place

4.2.3 Net Type and Color

Authors assign their own color to represent each template. As a default, the system provides colors for parallel, choice, and sequence library nets as shown in Table 8. There are several uses of template color. Figure 45 shows the nets and their net ids with their color at a selected region on the drawing canvas. When the search function finds nets, the nets are marked with dotted box according to their color as presented in Figure 46. Figure 47 shows the search result table that presents the net type with their color. Figure 48 shows net type with its coloring content layout. Also, a representative place uses net color to represent a collapsed net. The next section shows this example.

Table 8: Net Type and Its Color

Net Type	Color
(Parallel Template)	Cyan
(Choice Template)	Green
(Sequence Template)	Pink

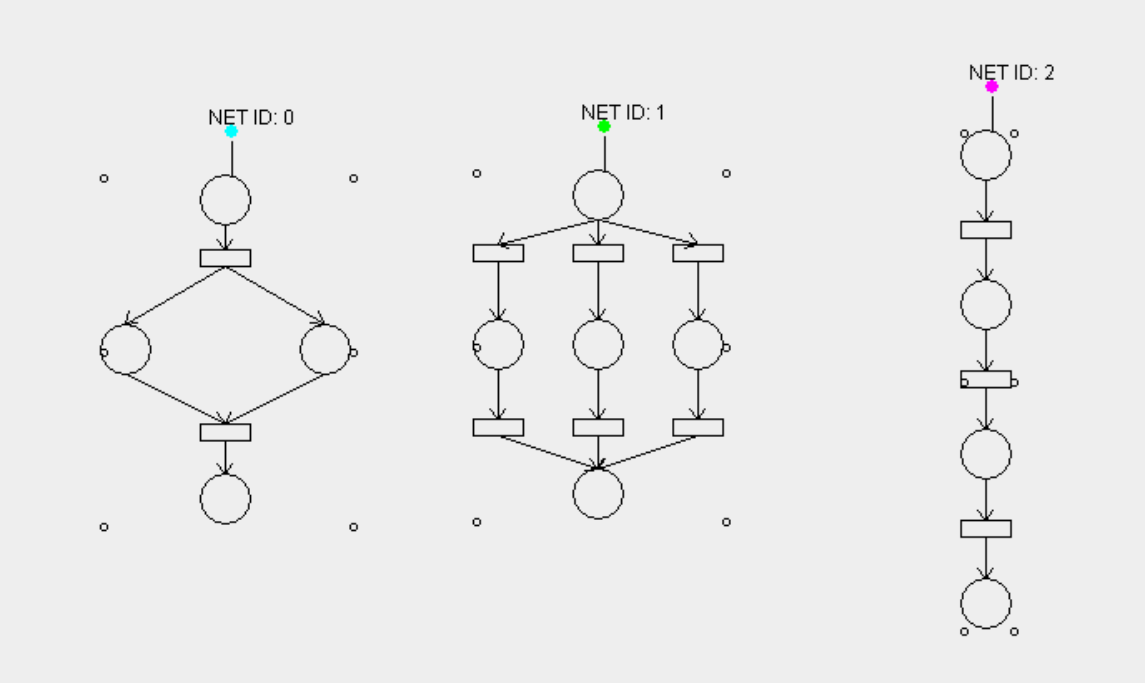


Figure 45: Net and Net ID with its Color at a Selected Region

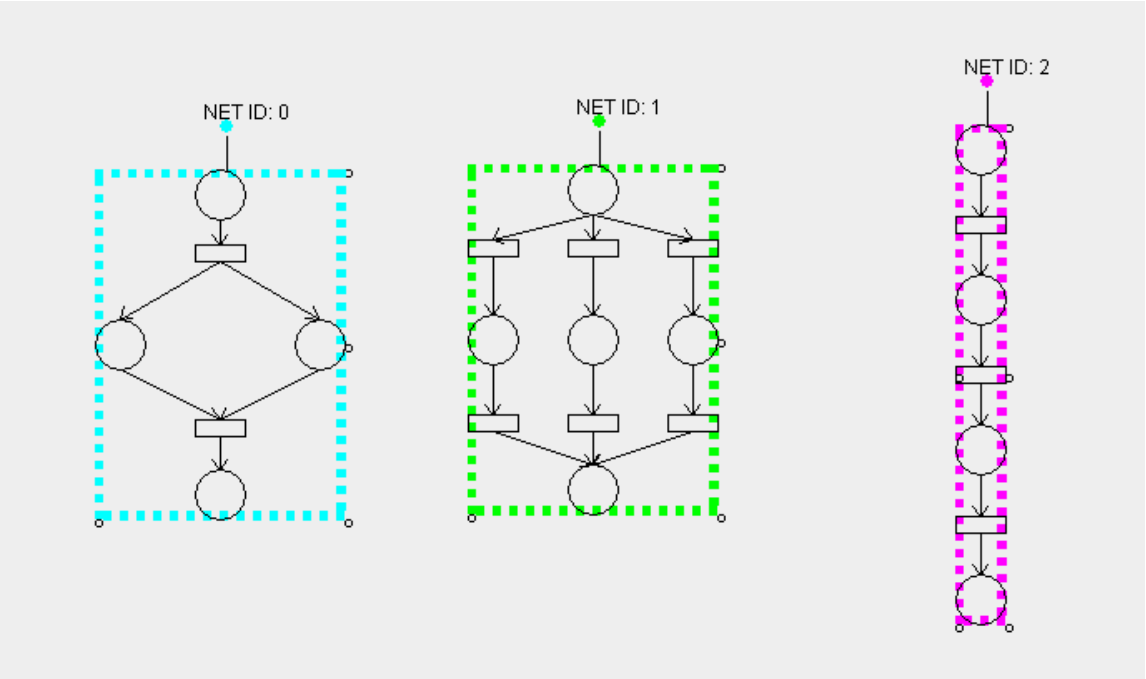


Figure 46: Search Result with Net Color

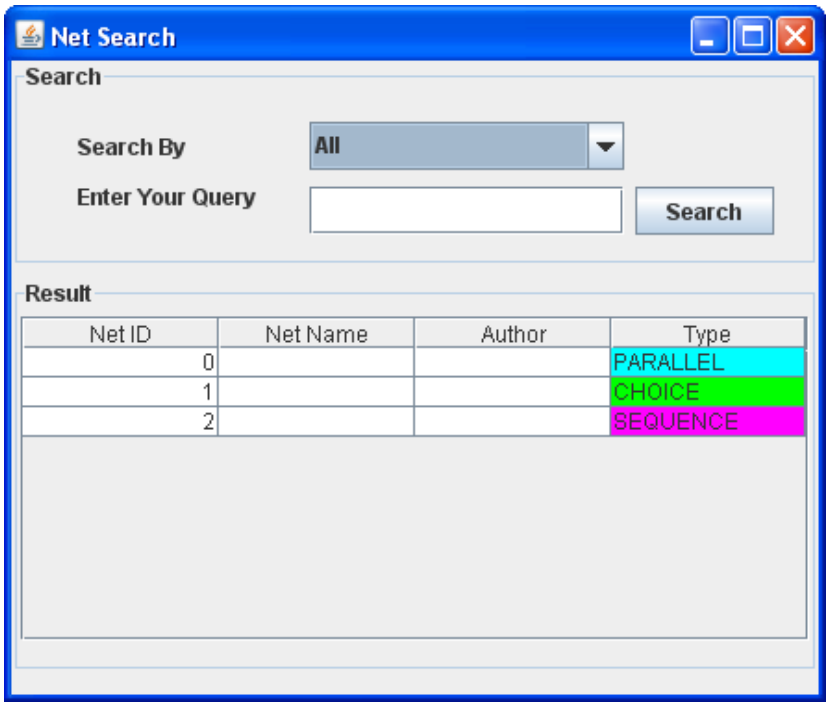


Figure 47: Net Type and Net Color at Search Table

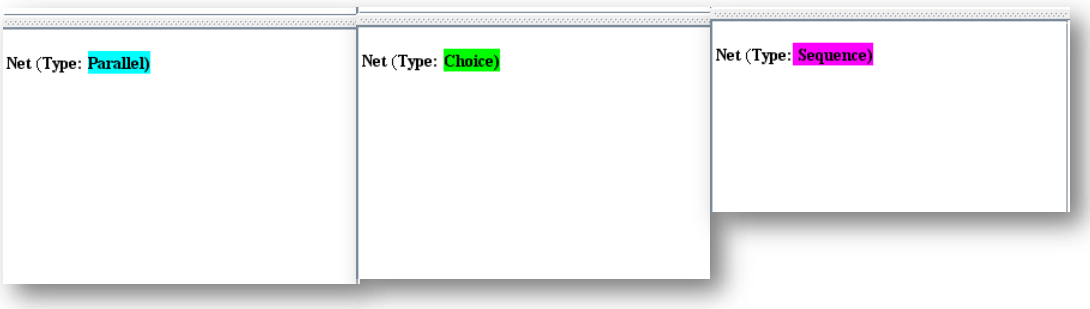


Figure 48: Net Type with its Color at Content Layout

4.2.4 Representative Place

TcAT provides a representative place that represents a collapsed net for the abstract view. Authors collapse a selected net by choosing the "collapse" item in a popup menu. The tooltip of a representative place provides brief information about the net including a captured image of the collapsed net. Figure 49 shows the representative place and its tooltip. As mentioned in previous section, the representative place and its tooltip use its color for net type.

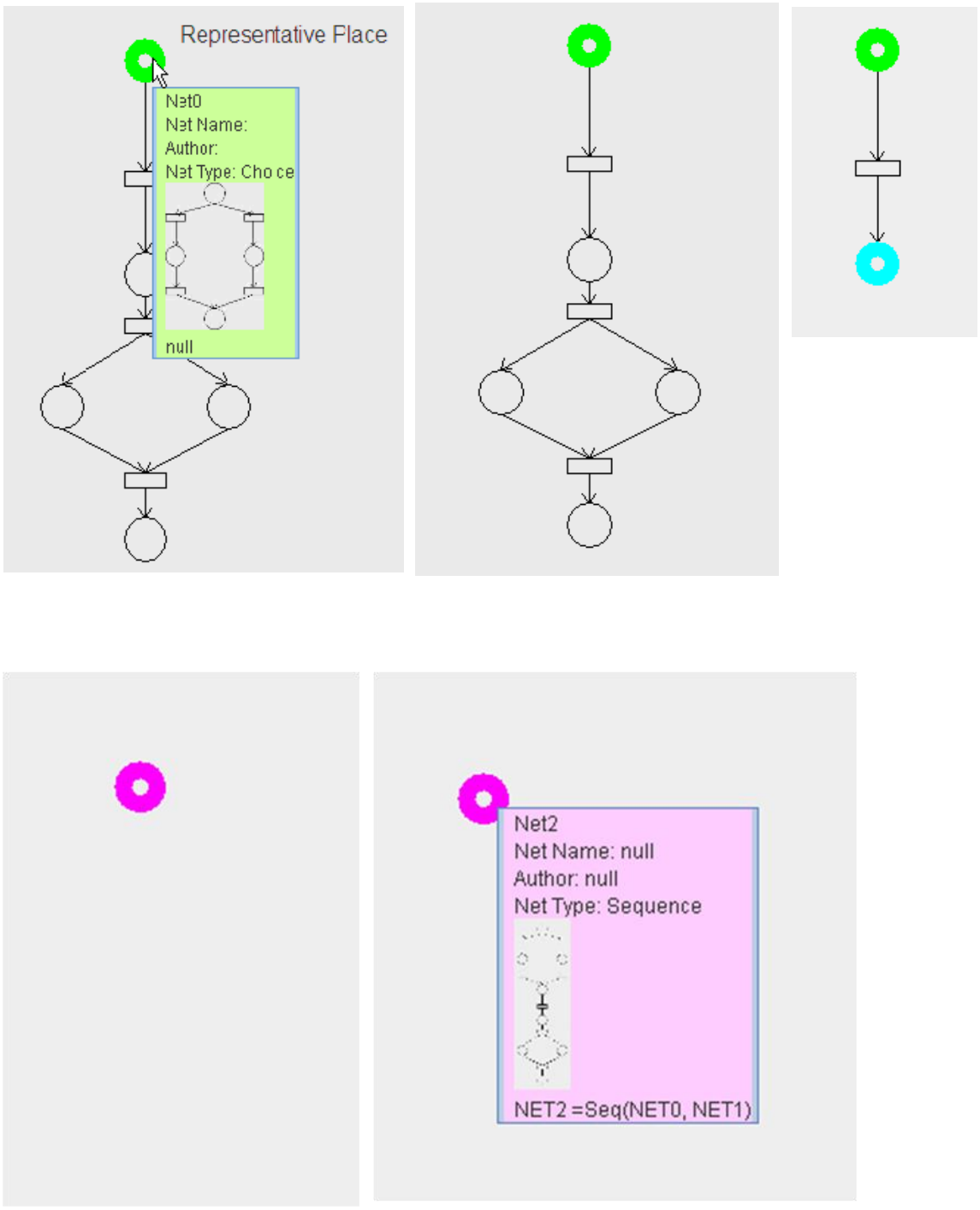


Figure 49: Representative Place and its Tooltip

4.2.5 Zoom In and Zoom Out

TcAT provides zoom in and zoom out views for drawing objects. Figure 50 shows a zoom out view and Figure 51 shows a zoom in view.

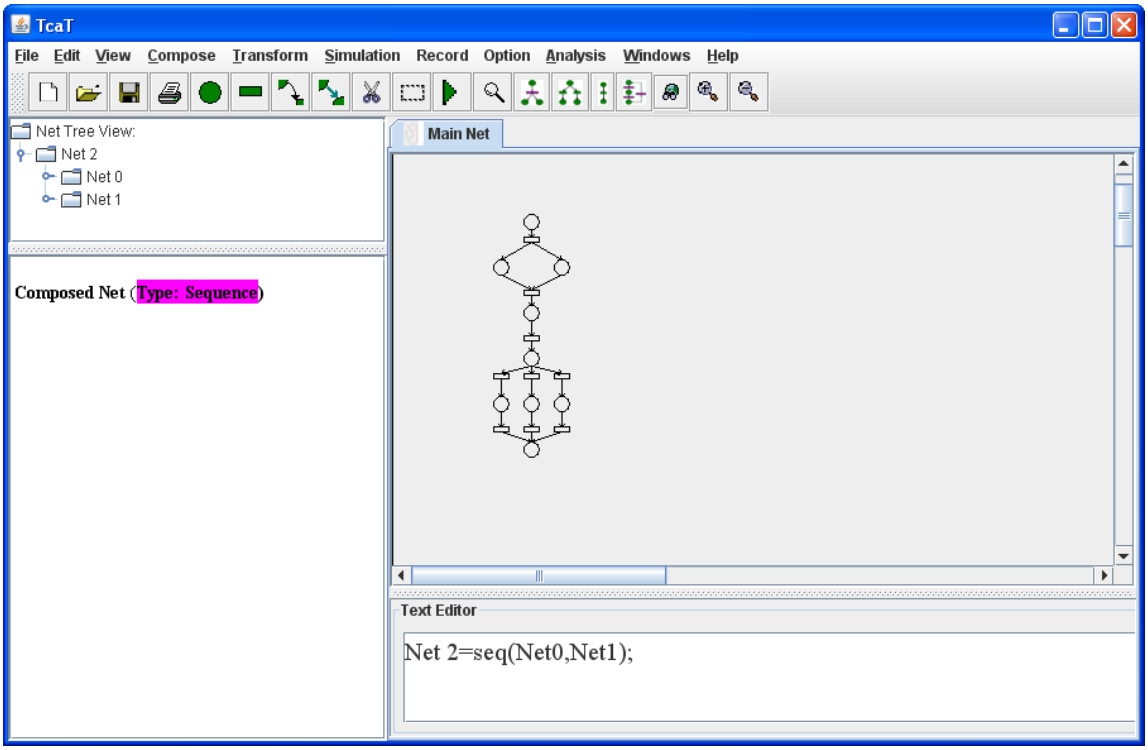


Figure 50: Zoom Out

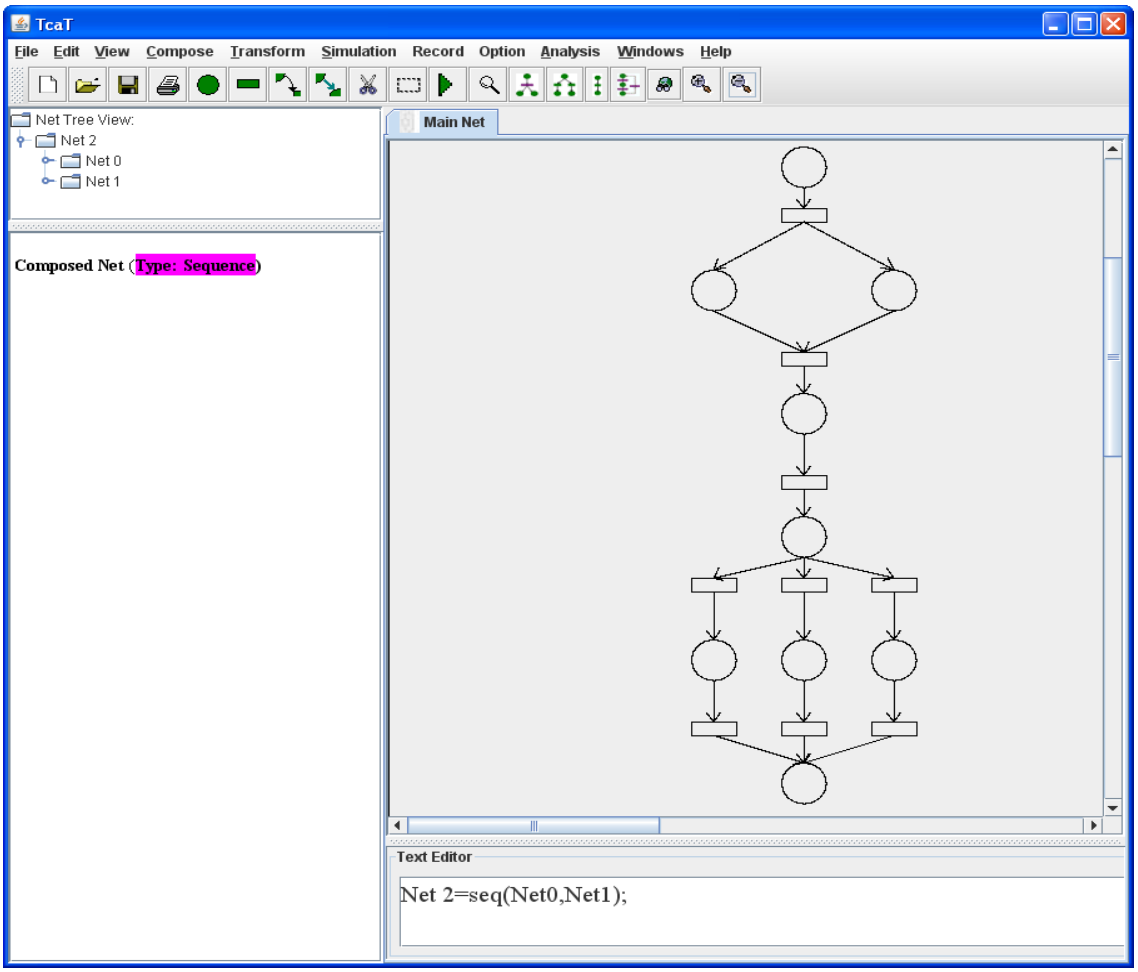


Figure 51: Zoom In

4.3 Implementation Details

This section provides implementation details to show how specific features are implemented.

4.3.1 How to Compute Position of Net Component

4.3.1.1 Automatic Net Creation

Figure 52 shows choice templates that have a different number of internal places—two, three, and four internal places. Figure 53 presents code for computing the positions of internal places and internal transitions of the choice template. The x position of each internal place (IP) is computed by the number of internal places of a template. The internal places are located evenly from the x position of the input place. The variable "spaceForXpos" represents the space between the x positions of places. The default value of this variable is 2× diameter of place. If the number of the internal places is odd number like Figure 52(b), one place is located at the middle among internal places. The y positions of all internal places are same. These are located as much as "spaceForYpos" (default value is 3×diameter of place) away from the input place's y position.

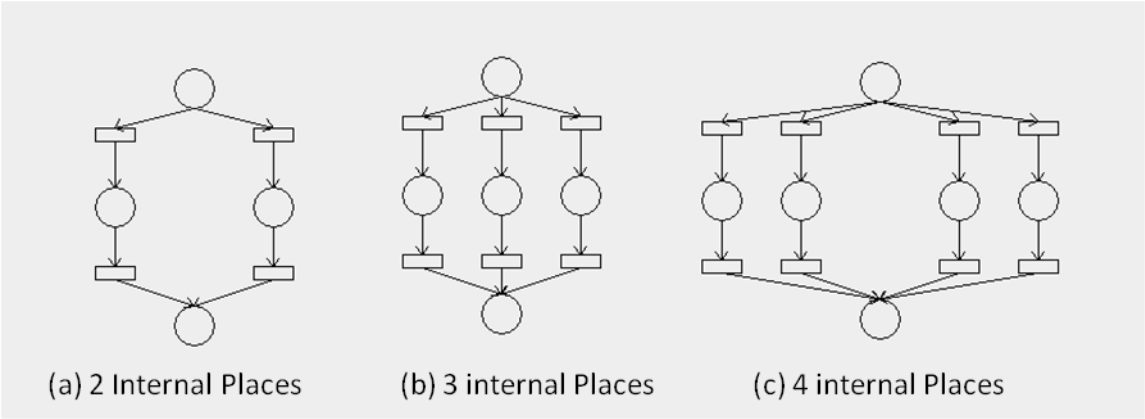


Figure 52: Automatic Net Creation: Choice Template

```
for (int j = 0; j < NofInterNode / 2; j++) {
    placeButton internodepb = new placeButton(drawPanel.NofPb++);
    xpos = InputPb.xpos - ( (j + 1) * spaceForXpos);
    ypos = InputPb.ypos + spaceForYpos;
    internodepb.setPosition(xpos, ypos);
    transButton preinternodetb = new transButton(drawPanel.NofTb++);
    xpos = internodepb.xpos;
     ypos = (InputPb.ypos + internodepb.ypos) / 2;
    preinternodetb.setPosition(xpos, ypos);
    transButton postinternodetb = new transButton(drawPanel.NofTb++);
    xpos = internodepb.xpos;
    ypos = (OutputPb.ypos + internodepb.ypos) / 2;
    postinternodetb.setPosition(xpos, ypos);
}
for (int j = 0; j < NofInterNode / 2; j++) {
    placeButton internodepb = new placeButton(drawPanel.NofPb++);
    xpos = InputPb.xpos + ( (j + 1) * spaceForXpos);
    ypos = InputPb.ypos + spaceForYpos;
    internodepb.setPosition(xpos, ypos);

    transButton preinternodetb = new transButton(drawPanel.NofTb++);
    xpos = internodepb.xpos;
    ypos = ( InputPb.ypos + internodepb.ypos) / 2;
    preinternodetb.setPosition(xpos, ypos);

    transButton postinternodetb = new transButton(drawPanel.NofTb++);
    xpos = internodepb.xpos;
    ypos = ( OutputPb.ypos + internodepb.ypos) / 2;
    postinternodetb.setPosition(xpos, ypos);
}
if (NofInterNode % 2 == 1) { // the number of Internal Place is odd number
    placeButton internodepb = new placeButton(drawPanel.NofPb++);
    xpos = InputPb.xpos;
    ypos = InputPb.ypos + spaceForYpos;
    internodepb.setPosition(xpos, ypos);

    transButton preinternodetb = new transButton(drawPanel.NofTb++);
    xpos = internodepb.xpos;
    ypos = ( (InputPb.ypos + internodepb.ypos) / 2);
    preinternodetb.setPosition(xpos, ypos);

    transButton postinternodetb = new transButton(drawPanel.NofTb++);
    xpos = internodepb.xpos;
    ypos = ( (OutputPb.ypos + internodepb.ypos) / 2);
    postinternodetb.setPosition(xpos, ypos);
}
```

Figure 53: Code for Creating Internal Places and Transitions for Choice Templates

The x position of a previous transition (•IP) and post transition (IP•) of each IP are the same as their IP. The y position of •IP is the middle point between the input place's y position and IP's y position. The y position of IP• is the middle point between IP's y position and the output place's y position.

In case of a parallel template, there is only one previous and post transition for all internal places. Figure 54 shows parallel templates having two, three, and four internal places. The positions of internal places of parallel template are same as choice templates. The x positions of two transitions (•IP, IP•) are the same as the x position of the input place. The y position of •IP is the middle point between the input place's y position and IP's y position. The y position of IP• is the middle point between IP's y position and the output place's y position. Figure 55 presents code for computing the positions of internal places and transition for the parallel template.

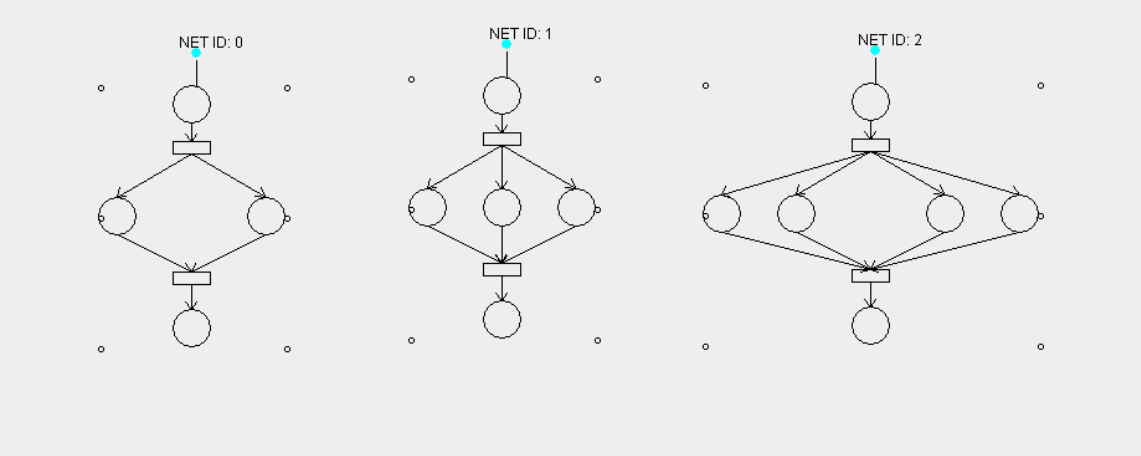


Figure 54: Automatic Net Creation: Parallel Template

```
for (int j = 0; j < NofInterNode / 2; j++) {
    placeButton internodepb = new placeButton(drawPanel.NofPb++);
    xpos = InputPb.xpos - ( (j + 1) * spaceForXpos);
    ypos = InputPb.ypos + spaceForYpos;
    internodepb.setPosition(xpos, ypos);
}

for (int j = 0; j < NofInterNode / 2; j++) {
    placeButton internodepb = new placeButton(drawPanel.NofPb++);
    xpos = InputPb.xpos + ( (j + 1) * spaceForXpos);
    ypos = InputPb.ypos + spaceForYpos;
    internodepb.setPosition(xpos, ypos);
}

if (NofInterNode % 2 == 1) { // the number of Internal Place is odd number
    placeButton internodepb = new placeButton(drawPanel.NofPb++);
    xpos = InputPb.xpos;
    ypos = InputPb.ypos + spaceForYpos;
    internodepb.setPosition(xpos, ypos);
}
yposofIPs = InputPb.ypos + spaceForYpos;

transButton preinternodetb = new transButton(drawPanel.NofTb++);
xpos = InputPb.xpos;
ypos = ( (InputPb.ypos + yposofIPs ) / 2);
preinternodetb setPosition(xpos, ypos);

transButton postinternodetb = new transButton(drawPanel.NofTb++);
xpos = InputPb.xpos;
ypos =( yposofIPs + placeButton.size + OutputPb.ypos) / 2;
postinternodetb setPosition(xpos, ypos);
```

Figure 55: Code for Creating Internal Places and Transitions for Parallel Templates

In case of a sequence template, the x position of each place and transition is the same as the input place. The y position of each transition is located as much as 2×diameter of place away from the previous place and the y position of each place is

located as much as “spaceYposforTtoPseq” (height of transition + diameter of place) away from the previous transition’s y position. Figure 56 shows sequence templates having one, two, and three internal places and Figure 57 presents code for computing the positions of internal places and transitions for the sequence template.

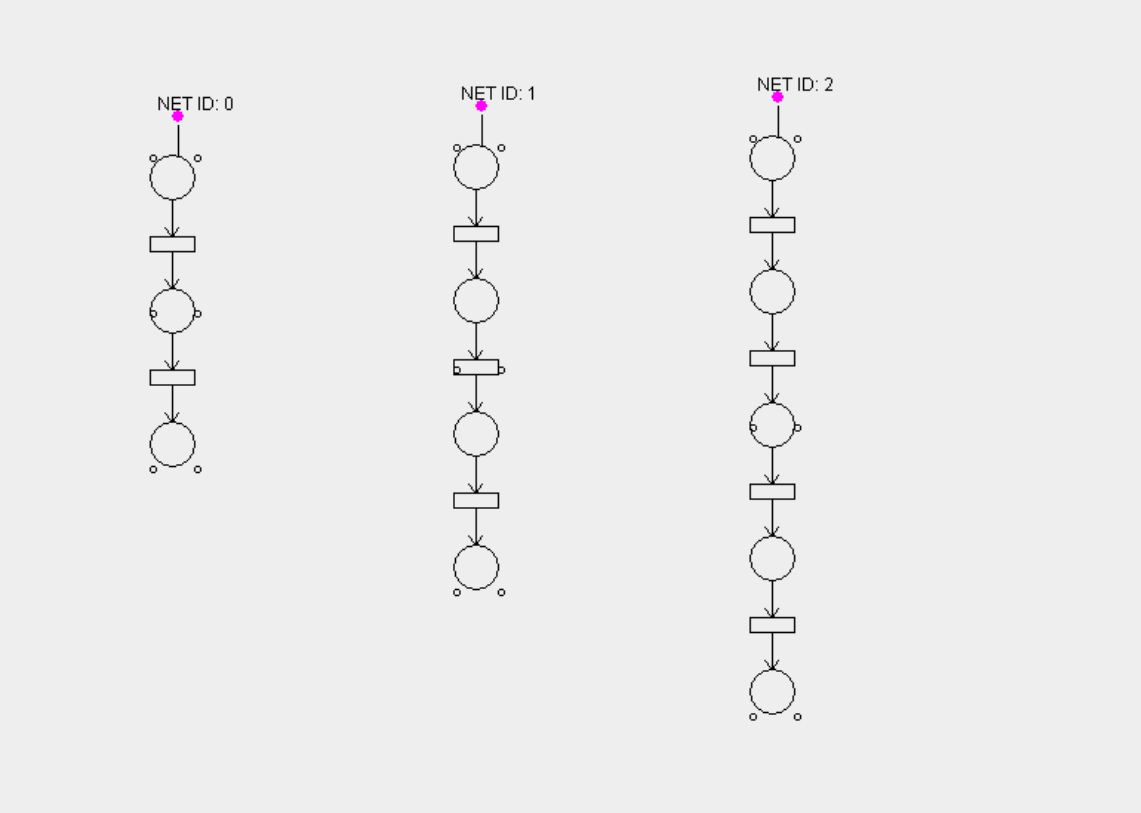


Figure 56: Automatic Net Creation: Sequence Template

```
spaceYposforPtoTseq=2*placeButton.size;
spaceYposforTtoPseq= transButton.height+ placeButton.size;
prevPb =InputPb;

for (int j = 0; j < NofInterNode; j++) {
    transxpos = prevPb.xpos;
    transypos = prevPb.ypos+ spaceYposforPtoTseq;
    transButton internodetb = new transButton(drawPanel.NofTb++);
    internodetb.setPosition(transxpos, transypos);

    placexpos = internodetb;
    placeypos = internodetb .ypos+ spaceYposforTtoPseq;
    placeButton internodepb = new placeButton(drawPanel.NofPb++);
    internodepb.setPosition(placexpos, placeypos);
    prevPb = internodepb;
}
```

Figure 57: Code for Creating Internal Places and Transitions for Sequence Templates

4.3.1.2 Net Composition

For automatic net composition by using choice operations, input place (I), output place (O), t_{I1} t_{I2}, t_{O1} and t_{O2} are automatically created. Figure 58 shows automatic net composition by combining subnets net_1 and net_2 with the choice operation.

The x position of t_{I1} is the same as the x position of I_1 (Input place of Net_1) and the y position of t_{I1} is the y position of I_1 − (2×the diameter of place). The x position of t_{I2} is the same as the x position of I_2 (Input place of Net_2) and the y position of t_{I2} is the y position of I_2 − (2×the diameter of place). The x position of t_{O1} is the same as the x position of O_1 and the y position of t_{O1} is the y position of O_1+ (2×the diameter of place). The x position of t_{O2} is the same as the x position of O_2 and the y position of t_{O2} is the y position of O_2+ (2×the diameter of place).

The x position of I of the composed net is the midpoint of the x position of I_1 and I_2. The y position of I is located with reference to the location that has the smaller value for the y position among t_1 and t_2. The x position of O is the midpoint of the x position of O_1 and O_2. The y position of O is located with reference to the location that has the larger value for the y position among t_1 and t_2. Figure 59 presents code for computing the locations for automatic choice composition.

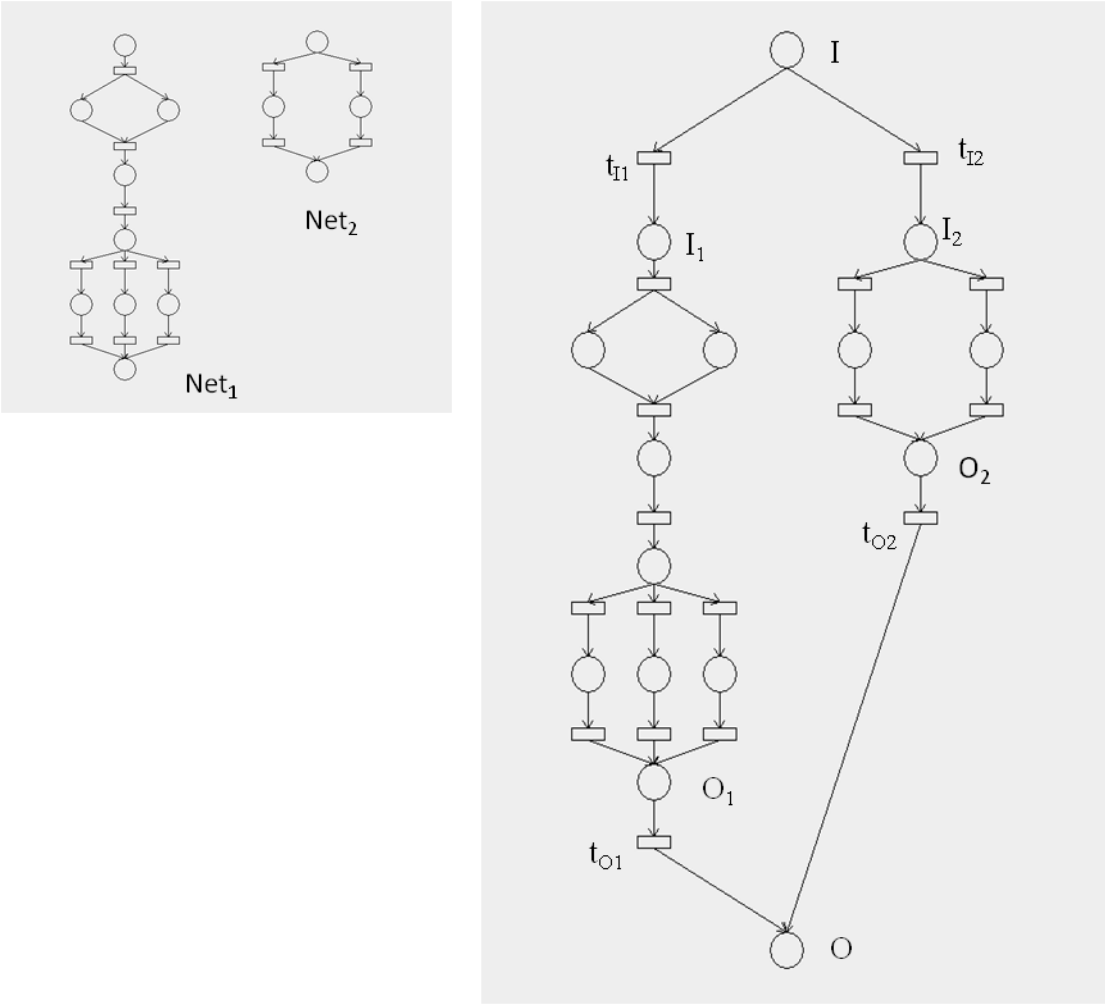


Figure 58: Automatic Net Composition with Choice: choice (Net_1, Net_2)

```
transtI1.xpos = t1.InputPb.xpos;
transtI1.ypos = t1.InputPb.ypos – 2*placeButton.size;
transtI2.xpos = t2.InputPb.xpos;
transtI2.ypos = t2.InputPb.ypos - 2*placeButton.size;
transtO1.xpos= t1.OutputPb.xpos;
transtO1.ypos =t1.OutputPb.ypos + 2*placeButton.size;
transtO2.xpos = t2.OutputPb.xpos;
transtO2.ypos = t2.OutputPb.ypos + 2*placeButton.size;

InputPb.xpos = (t1.InputPb.xpos+ t2.InputPb.xpos)/2;
if (t1.InputPb.ypos > t2.InputPb.ypos)
    InputPb.ypos=t2.InputPb.ypos-2*placeButton.size;
else
    InputPb.ypos=t1.InputPb.ypos-2*placeButton.size;

OutputPb.xpos = (t1.OutputPb.xpos+ t2.OutputPb.xpos)/2;
if (t1.OutputPb.ypos > t2.OutputPb.ypos)
    OutputPb.ypos=t1.OutputPb.ypos+2*placeButton.size;
else
    OutputPb.ypos=t2. OutputPb.ypos+2*placeButton.size;
```

Figure 59: Code for Computing Positions for Choice Composition

For parallel net composition, input place (I), output place (O), t_I and t_O are automatically created. The x position of t_I is the midpoint of the x position of I_1 and I_2. The y position of t_I is located with reference to the location that has the smaller value for the y position among I_1 and I_2. The x position of t_O is the midpoint of the x position of O_1 and O_2. The y position t_O is located with reference to the location that has the larger value for the y position among O_1 and O_2. I's x position is the same as the x position of t_I and I's y position is t_I's y position − (2×the diameter of place). O's x position is the same with x position of t_O and O's y position is t_O's y position + (2×the diameter of place).

For sequence net composition for net_1 and net_2, seq(Net_1, Net_2), connection transition t is automatically created. The x position of t is same as the x position of O_1 (net_1's output place) and the y position of t is the y position of O_1 + (2×the diameter of place). Since I_2 (net_2's input place) follows t, the new y position of I_2 is the y position of t + the height of the transition + diameter of the place. The locations of all elements of net_2 are moved according to the difference between the previous position and the new position of I_2. Figure 60 presents code for computing the locations for automatic sequence composition. Figure 61 shows automatic net composition by combining subnets net_1 and net_2 with the sequence operation.

```
transxpos = net1.OutputPb.xpos;
transypos = net1.OutputPb.ypos + 2*placeButton.size;
transButton t = new transButton(drawPanel.NofTb++);
t.setPosition(transxpos, transypos);

int xdiff =t.xpos -net2.InputPb.xpos ;
int ydiff =t.ypos+ transButton.height+placeButton.size )-net2.InputPb.ypos;
for(int i=0;i<net2.PlaceList.size();i++)
{
    placeButton nt = (placeButton)net2.PlaceList.elementAt(i);
    nt.setPosition(nt.xpos+xdiff, nt.ypos+ydiff);
}
for(int i=0;i<net2.TransList.size();i++)
{
    transButton nt= (transButton)net2.TransList.elementAt(i);
    nt.setPosition(nt.xpos+xdiff, nt.ypos+ydiff);
}
for(int i=0;i<net2.ArcList.size();i++)
{
    arc arct = (arc)net2.ArcList.elementAt(i);
    arct.setfromPosition(arct.fromxpos+xdiff, arct.fromypos+ydiff);
    arct.settoPosition(arct.toxpos+xdiff, arct.toypos+ydiff);
}
```

Figure 60: Code for Computing Positions for Sequence Composition

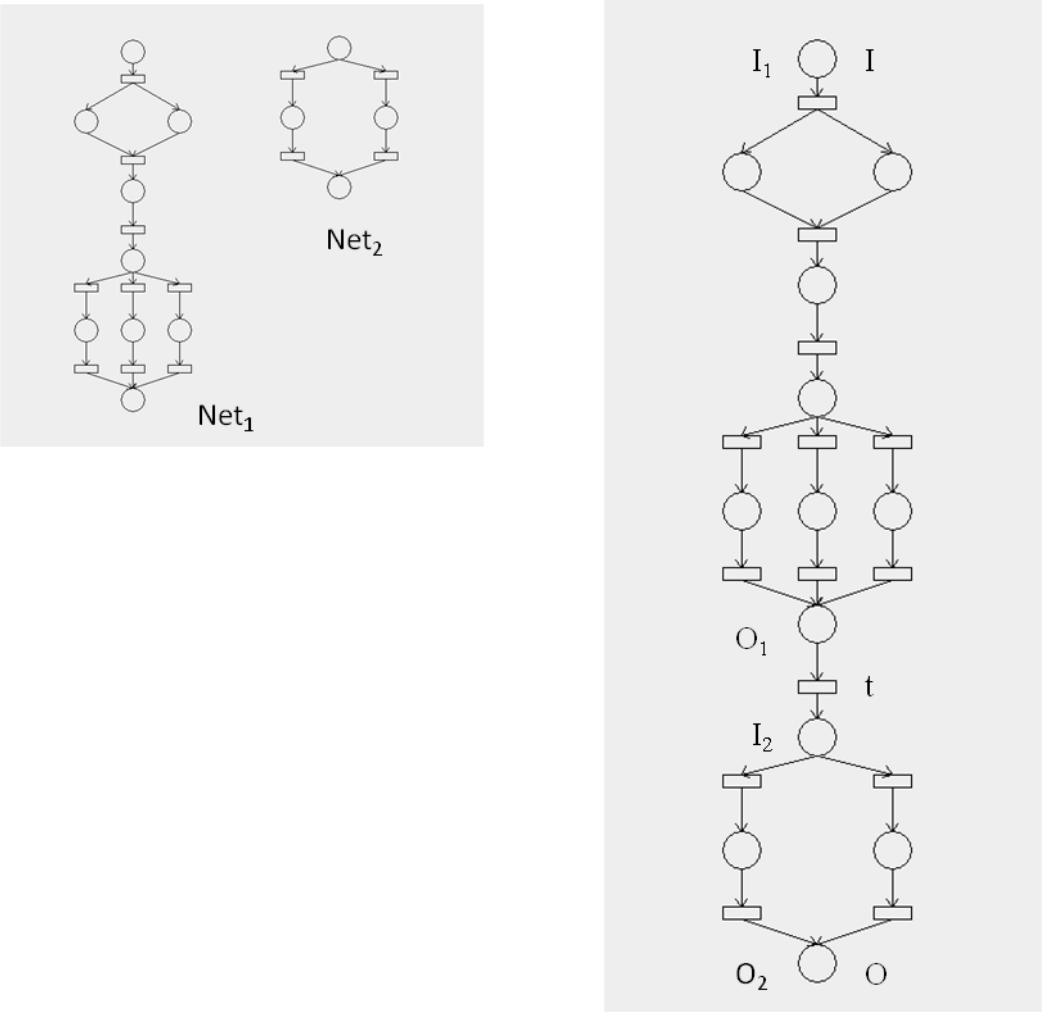


Figure 61: Automatic Net Composition with Sequence: seq (Net_1, Net_2)

CHAPTER V

USABILITY EVALUATION

The chapter describes a usability evaluation of the proposed authoring prototype for large and complex hypertext with reusable components. How effectively subjects can author large and complex hypertext using this prototype was mainly evaluated through this user evaluation. Since this research involves human subjects, IRB (Institutional Review Board) approval was received through the application review process.

This chapter provides the goal of the study and the design of the study including subjects, tasks, and experimental methods. After quantitative/numeric data and qualitative/observational data presentation, analysis and interpretation of the data follows.

5.1 Goal of the Study

The purpose of this study is to verify the proposed features are useful for authoring large and complex hypertext in a Petri net-based hypertext system. We want to show how much improvement is made when we compared the previous system and the prototype. For this comparative analysis, we evaluated the previous caT authoring tool, xTed and the prototype, TcAT. For TcAT, there are two interaction methods to compare: graphical interaction and textual language.

Also HTML-based authoring (either directly edited HTML or editing with an HTML authoring tool such as FrontPage) was compared with Petri net-based authoring

to examine how subjects react to Petri net-based authoring for hypertext. This evaluation was executed only for experts who know HTML-based authoring.

In addition, this study investigated the following significant characteristics for authoring large and complex hypertext. These investigation results would be helpful information for future implementations for authoring of Petri net-based hypertext document systems.

- Features of the proposed tool that are helpful in creating large and complex hypertexts
- Features of the proposed tool that make it difficult to create large and complex hypertexts
- Features users like to add/delete to the proposed prototype
- Most difficult task to perform
- Least difficult task to perform

5.2 Design of the Study

5.2.1 Subjects

15 subjects were recruited by email and person-to-person contacts. We assigned a random identifier to each subject. Subjects were asked to fill out a demographic questionnaire, to perform specific assigned tasks, to answer the questions regarding each task after performing the task, to answer the summary questionnaire for this study, and to provide reasons for their answers, responses and behaviors. They were free to refuse to answer any of the questions that made them uncomfortable. They could quit anytime whenever the subject did not want to continue. They were not exposed to any risks in the

study. There was no immediate and direct benefit to them and no compensation for their participation.

Subjects were classified as novice, intermediate, and expert group based on their completions of the demographic questionnaire presented in Appendix C. That questionnaire includes information about the subject's experience with computers, programming languages, and using and drawing graphs and Petri nets. The expert subjects have some experiences using web programming, Petri nets, and graphs. The intermediate subjects have some computer science and engineering background including programming languages. The novice subjects can use computers and the internet well. We had four experts, six intermediate, and five novices.

All of the experts were computer science majors; three are Ph.D. students at Texas A&M University and one is a member of the staff of Texas A&M University and has a master’s degree. All of them have created web documents for more than 2 years and two of them know Petri nets. None of experts used Petri net editing tools or created hypertexts using a Petri net Authoring Tool before this evaluation. Only one expert has experience drawing a Petri net by hand. Table 9 presents the specific information about the expert subjects.

For intermediate subjects, five are computer science Ph.D. students of computer science and one is a researcher who has a Ph.D. degree of industrial engineering. All of them have a strong computer science background including programming languages. Since they hesitated to make a web-based hypertext document, even though many of

them have experience in creating web documents and knowledge about Petri nets as presented in Table 10, we categorized them as intermediate subjects.

Table 9: Profiles for Expert Subjects

Characteristics	Subject			
Random Number	E-1	E-2	E-3	E-4
How long authoring web pages	More than 2 years	More than 2 years	More than 2 years	More than 2 years
Know Petri net	yes	yes	no	no
Draw a Petri net	yes	no	no	no
Method for drawing a Petri net	By hand	N/A	N/A	N/A
Created hypertext using Petri net Authoring Tool	no	no	no	no

Table 10: Profiles for Intermediate Subjects

Characteristics	Subject					
Random Number	I-1	I-2	I-3	I-4	I-5	I-6
Major Field	CPSC	IE	CPSC	CPSC	CPSC	CPSC
How long authoring web pages	Less than 6 months	Less than 6 months	1 year to 2 years	Less than 6 months	More than 2 years	No experience
Know Petri net	yes	yes	yes	no	no	yes
Draw a Petri net	yes	yes	yes	no	no	no
Method for drawing a Petri net	By hand	By hand	By hand	N/A	N/A	N/A
Created hypertext using Petri net Authoring Tool	no	no	no	no	no	no

We classified subjects who have no computer science background as novice. Two of them are undergraduate students of biochemistry and business, one is a graduate student of education, one is a pastor with master's degree in theology and one is a house wife who has a bachelor's degree in engineering. All of them use computers and the Internet daily and have used them for more than 2 years. Instead of Petri nets, one subject, majoring in an education major, is familiar with graphs and their drawing and had experience using a graph editor tool. Table 11 shows profiles for novice subjects.

Table 11 Profiles for Novice Subjects

Characteristics	Subject				
Random Number	N-1	N-2	N-3	N-4	N-5
Major Field	Biochemistry	Education	Engineering	Business	Theology
How long use computer	More than 2 years	More than 2 years	More than 2 years	More than 2 years	More than 2 years
How long authoring web pages	Less than 6 months	No experience	No experience	No experience	No experience
Know Petri net	no	no	no	no	no
Draw a Petri net	no	no	no	no	no
Method for drawing a Petri net	N/A	N/A	N/A	N/A	N/A
Created hypertext using Petri net Authoring Tool	no	no	no	no	no

Novice and intermediate subjects performed the same tasks. We assigned extra work creating a web-based digital library to expert subjects to compare web-based and Petri net-based authoring. Figure 62 summarizes the goals.

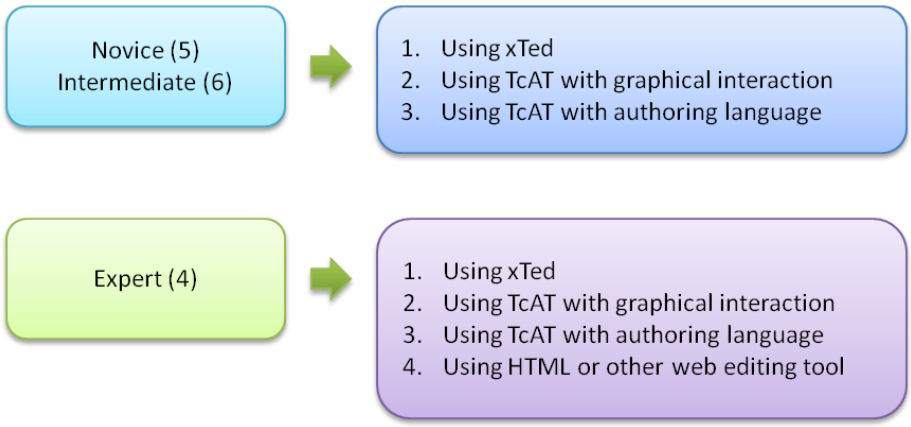


Figure 62: Subjects and Evaluation Target

5.2.2 Experimental Methodology

The study followed the procedure described in Figure 63. Before performing tasks, we gave extensive tutoring about xTed and TcAT to each subject. Including demonstrating the overview of the prototype system, we introduced the basic knowledge of Petri nets and its theories, Trellis, xTed, and TcAT. When a subject performed the tasks, we observed the subject and helped the subject when the subject had problem performing the tasks.

1. Introduce the study
    - describe the purpose of the study
    - describe the experimental protocol
2. Get user's consent for testing
    - if a user agrees to continue, get the informed consent and audio tape consent signed by the user and give a copy to the user
3. Allocate a random identifier to the user for confidentiality purposes for all documentation during and after the study
4. Complete the demographic questionnaire
    - this questionnaire includes information about the user's experience with using and drawing graph and Petri net.
    - users can be classified (novice, experienced, expert) based on their experience with using and drawing graphs and Petri nets
5. Tutor xTed and TcAT
6. Perform the task
    - the user may be observed
    - break anytime whenever the user does not want to continue
7. Complete the post-task questionnaire
    - this questionnaire asks users about the tasks, their thoughts and suggestions, etc.
8. Free-form interview
    - the test conductor may ask questions to users about what he/she observed and ask for general comments that the user may have. This interview may be audio taped if the user agrees.
9. Thank the user.

Figure 63: Procedure of the User Study

To verify the usefulness of proposed features, completeness (to measure whether each task is performed properly or not at a given time) and effectiveness (to measure whether authors can achieve task fast and easily) were used. Time taken for authoring is a major criterion to measure effectiveness.

Also, usefulness of the proposed features was verified by post-task questionnaires and interviewing. As post-task summary questionnaire, we have total of twenty six questions. For the fourteen questions among this questionnaire, subjects rated each question from 1 to 5 (1 is strongly disagree and 5 is strongly agree); those results are presented in the quantitative data section. Subjects wrote down their ideas for the other fourteen questions. Along with the result of free-form interviewing, those answers are presented at qualitative data section.

5.2.3 Tasks

After learning about Petri net, caT including xTed, and TcAT, subjects performed the task of creating a digital art museum by combining materials from digital collections for fine artists. This museum can be composed from individual collections of Vincent van Gogh and Claude Monet. We provided 17 paintings of van Gogh and 10 paintings of Monet. Each painting has an image file, a summary file that includes its title, category, date, exhibition location, and its relevant article files; see Figure 64. Subjects can organize the resources by their own policies and rules.

For comparative analysis, each subject performed the same task three times or four times by using different methods. The order of methods is fixed (1. xTed, 2. TcAT

with graphical interaction, 3 TcAT with textual language, 4 HTML-based authoring). In this study we did not consider the learning effects. We emulated xTed in a new tool to get a more realistic result by providing same environment with a TcAT.

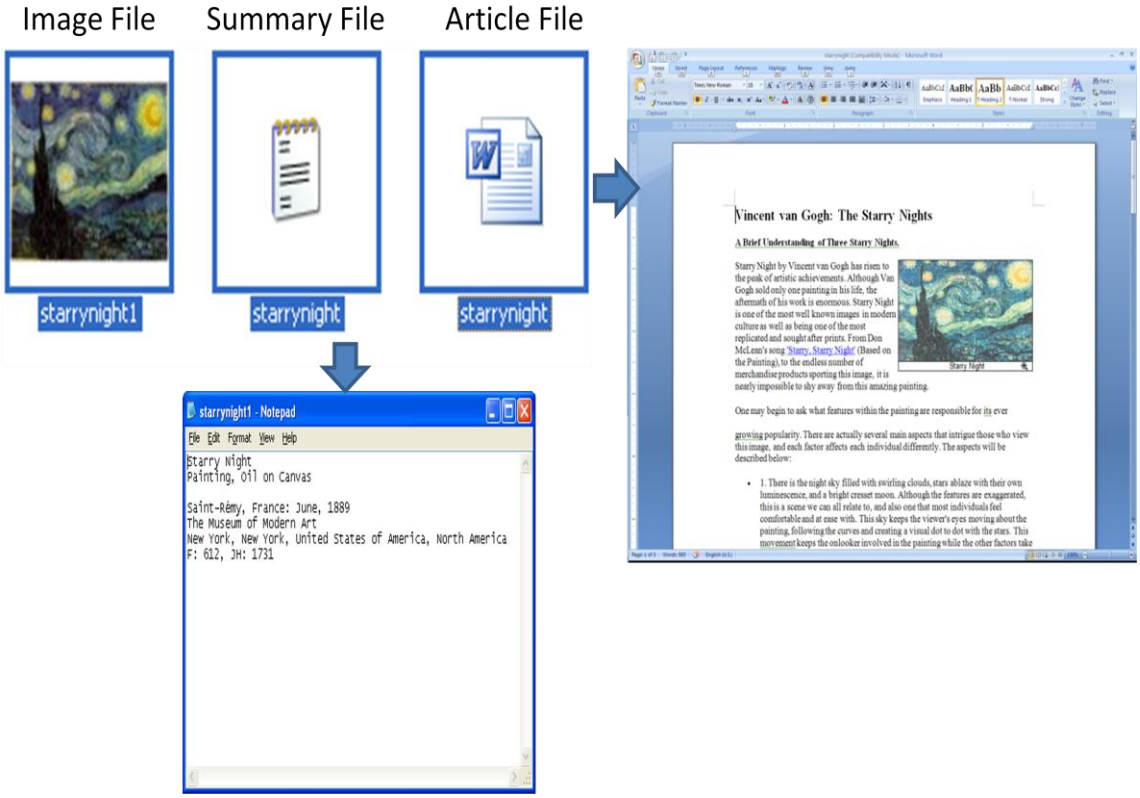


Figure 64: Resources of Painting

As an example of the task, Figure 65 shows a web-based digital museum of van Gogh and Monet created by a subject. This subject organized each collection by the years when the paintings were created. Figure 66 shows the result of xTed, Figure 67 shows result of TcAT with graphical interaction, and Figure 68 shows the result of TcAT with textual language.

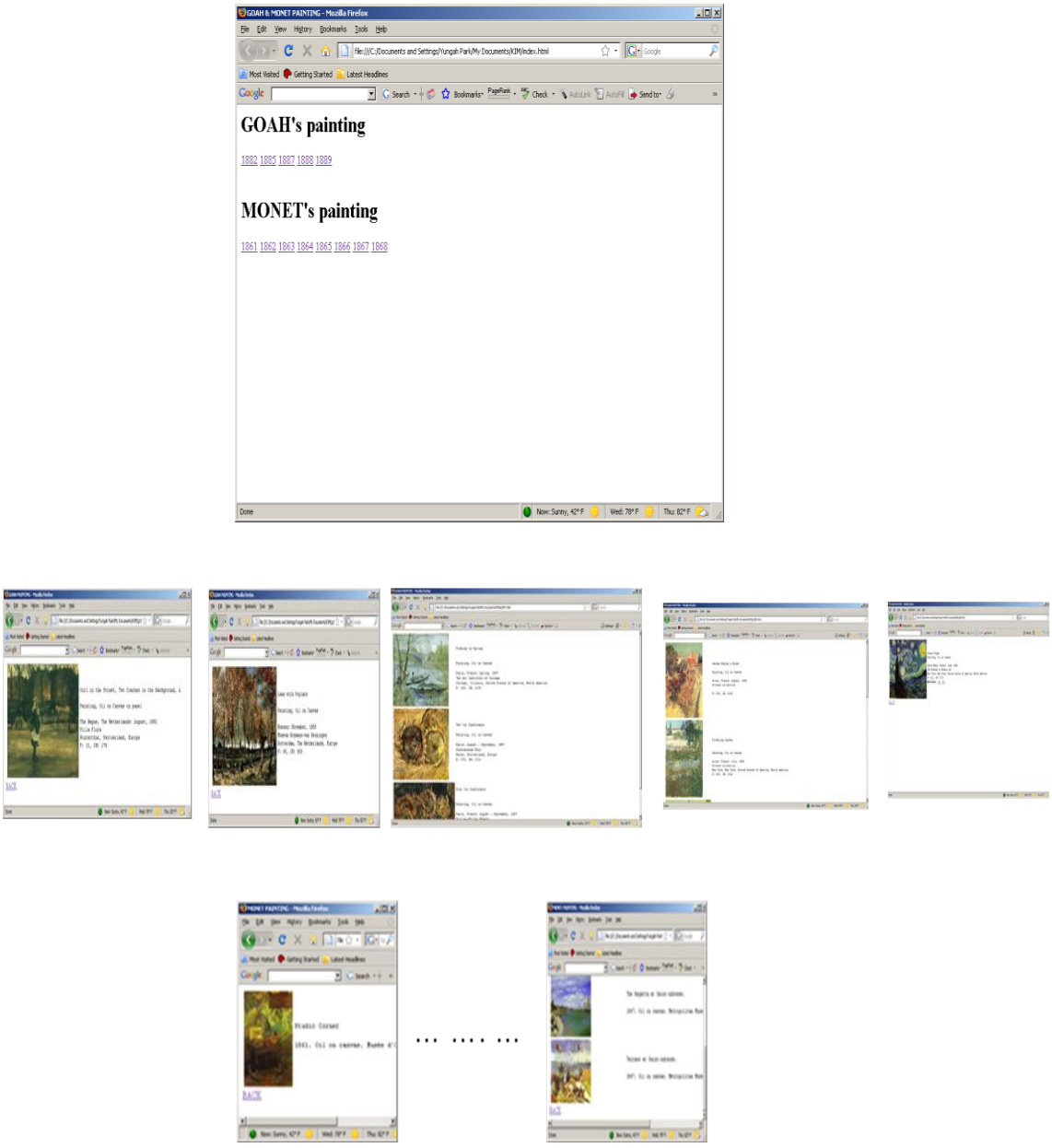


Figure 65: Web-based Digital Art Museum Created by a Subject

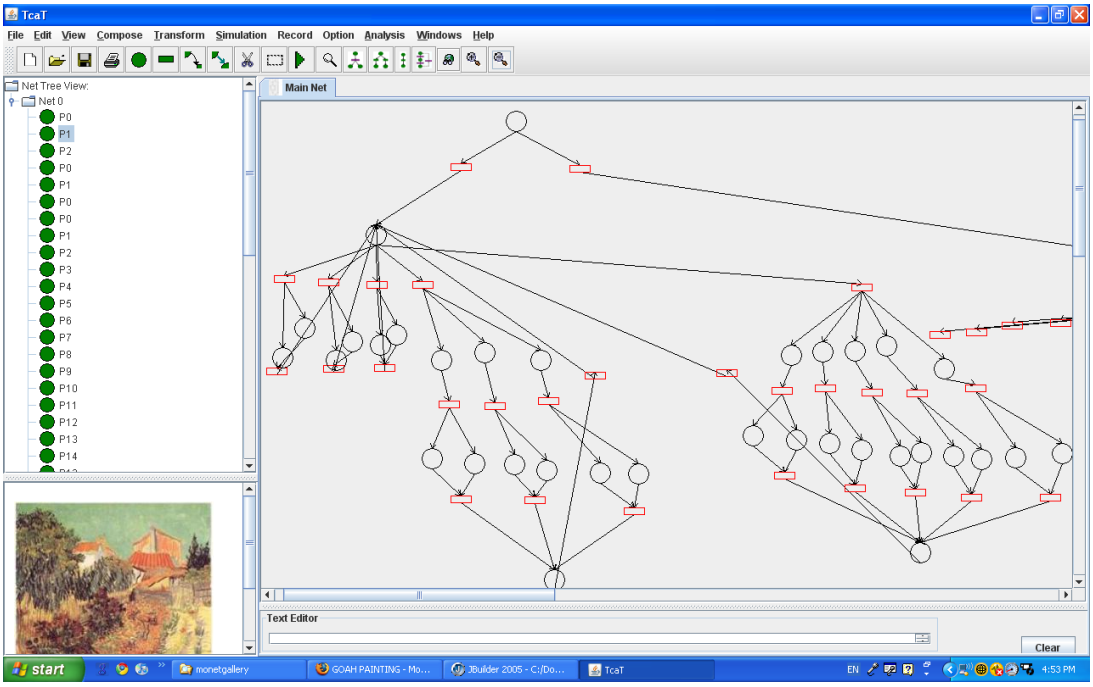


Figure 66: Result of using xTed

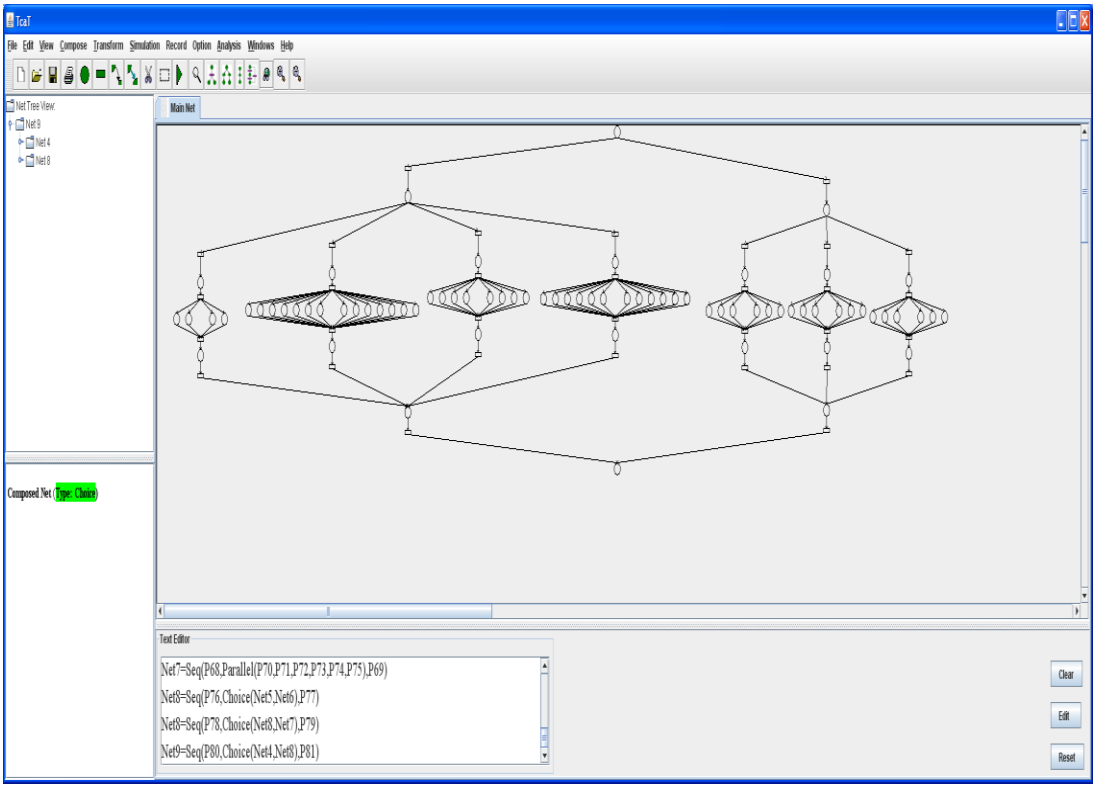


Figure 67: Result of using TcAT with Graphical Interaction

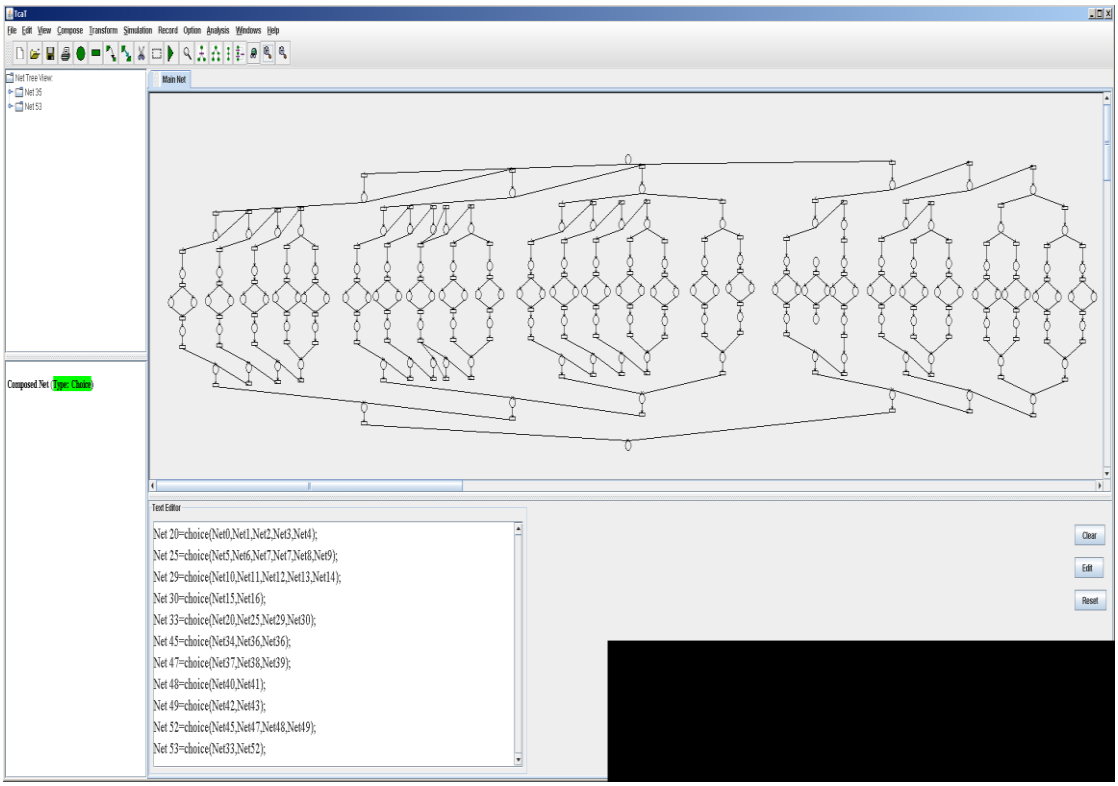


Figure 68: Result of TcAT with Textual Language

5.3 Quantitative Data

5.3.1 Time Taken to Perform Task

Time taken to make a combined digital art collection by every subject, which is the main criterion of measuring effectiveness as mentioned in the experimental methodology section, is presented in Figures 69 and 70. Figure 69 shows the whole data by each subject. Except for one case (novice 1) we can see that using textual language is the quickest method to make a digital library by each subject. Every subject took longer time with xTed than with TcAT with graphical interaction and TcAT with textual language. On the overall average it took 31.88 minutes. Novice 2 took the longest time

(79 minutes) to complete the task with xTed and intermediate 6 took shortest time (9 minutes) with textual language as a whole. Figure 70 shows the histogram that displays the distribution of a time data set.

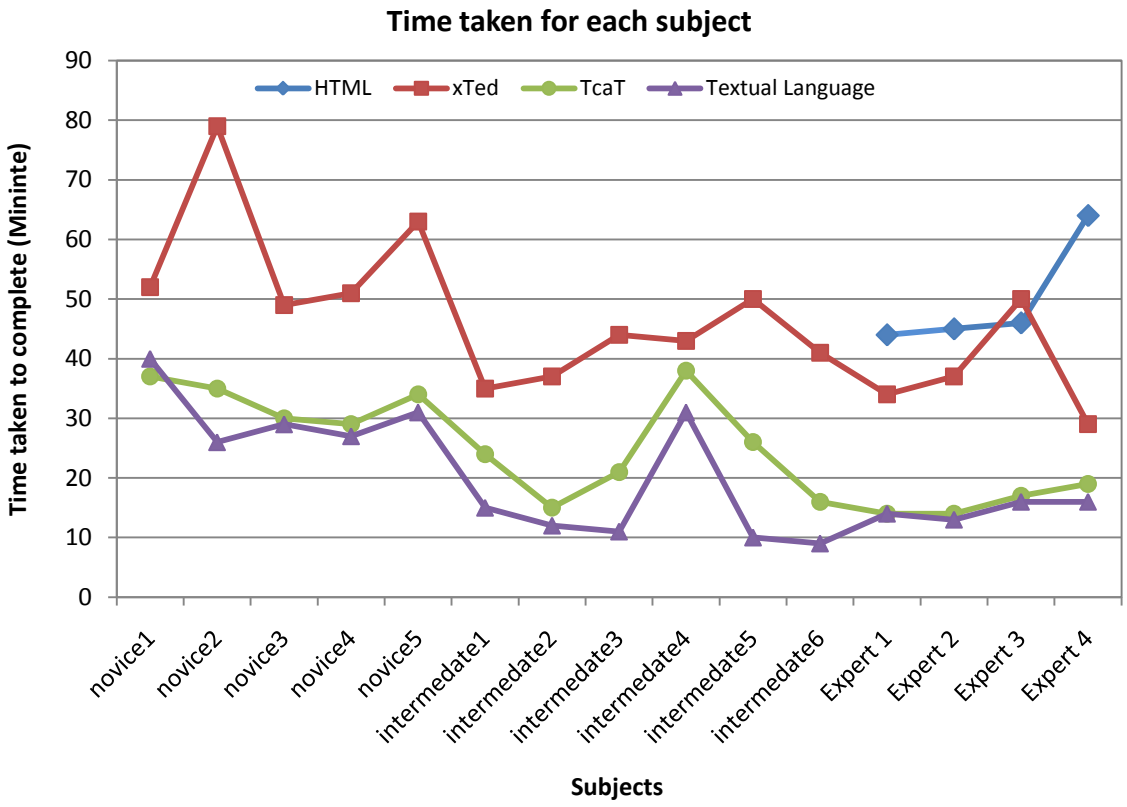


Figure 69: Time Taken to Make a Digital Art Collection

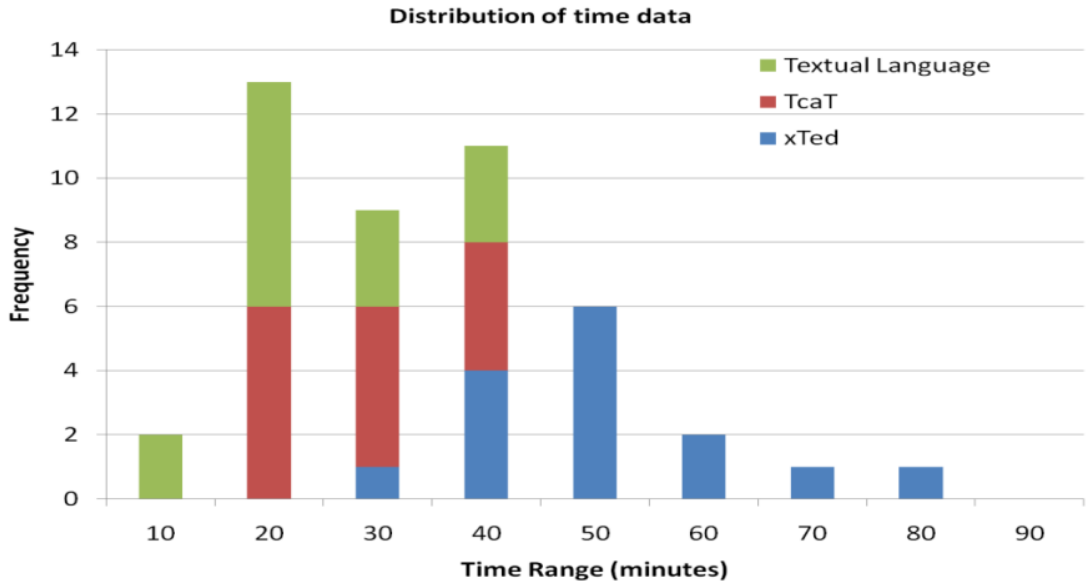
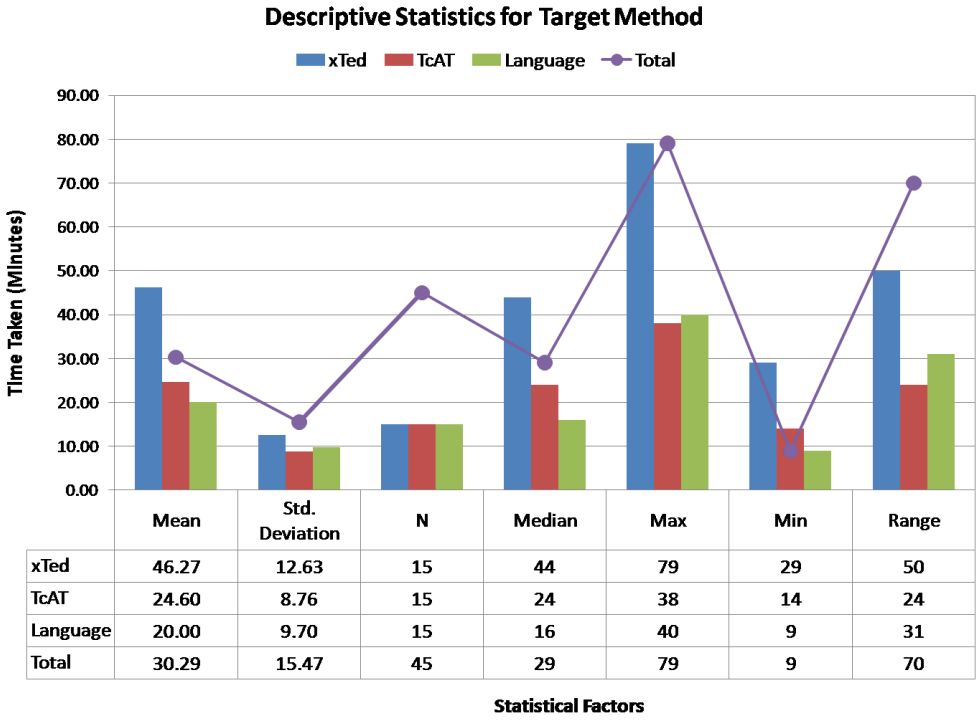


Figure 70: Histogram of Time Data

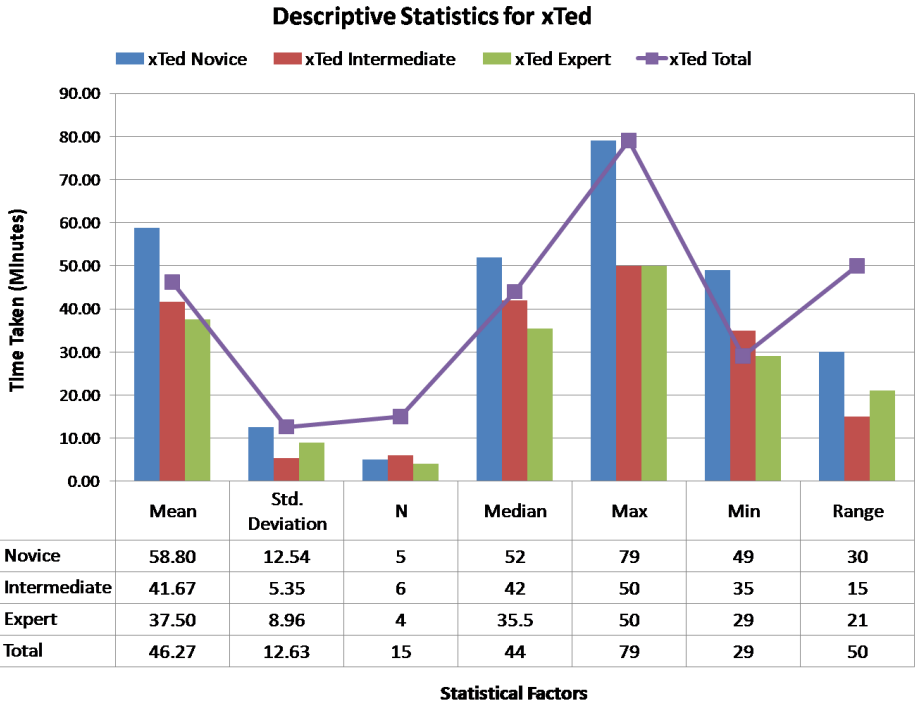
Figure 71 shows more detailed statistical information for three different target methods; xTed, TcAT with GI (i.e., Graphical Interaction) and TcAT with textual language. For average values, TcAT with textual language is the best, TcAT with GI is the second, and xTed is the worst performance. The variation in values with TcAT with GI is the slightest and with xTed is the most. For xTed, expert 4 showed the best performance (29 minutes) and novice 2 showed the worst performance (79 minutes). For TcAT GI, novice 1 gave the worst performance (38 minutes) and intermediate 2 showed the best performance (14 minutes). For TcAT textual language, novice 1 showed the worst performance (40 minutes) and intermediate 6 showed the best performance (9 minutes).



	Mean	Std. Deviation	N	Median	Max	Min	Range
xTed	46.27	12.63	15	44	79	29	50
TcAT	24.60	8.76	15	24	38	14	24
Language	20.00	9.70	15	16	40	9	31
Total	30.29	15.47	45	29	79	9	70

Figure 71: Descriptive Statistics for Target Method

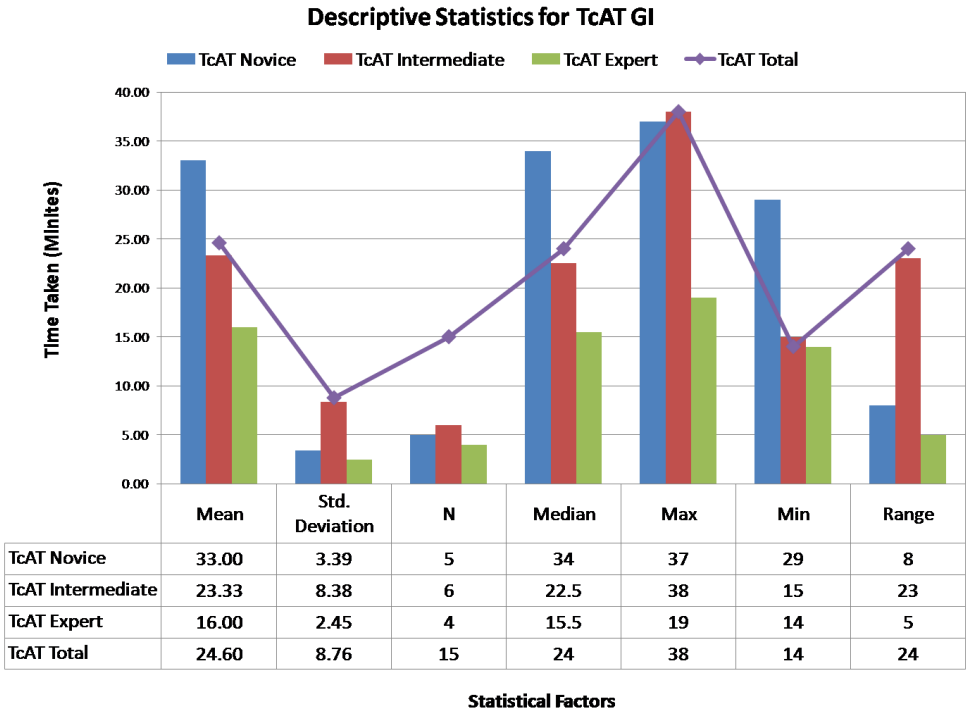
Figure 72, 73, and 74 show descriptive statistical information when using three different target methods by each subject type. For xTed, as shown in Figure 72, for average values, the novice group made the worst and expert group the best performance. The variation in values by the intermediate group is the slightest and by novice is the most. The minimum value by the novice group is similar to the maximum values by the intermediate and the expert group.



	Mean	Std. Deviation	N	Median	Max	Min	Range
xTed Novice	58.80	12.54	5	52	79	49	30
xTed Intermediate	41.67	5.35	6	42	50	35	15
xTed Expert	37.50	8.96	4	35.5	50	29	21
xTed Total	46.27	12.63	15	44	79	29	50

Figure 72: Descriptive Statistics for xTed

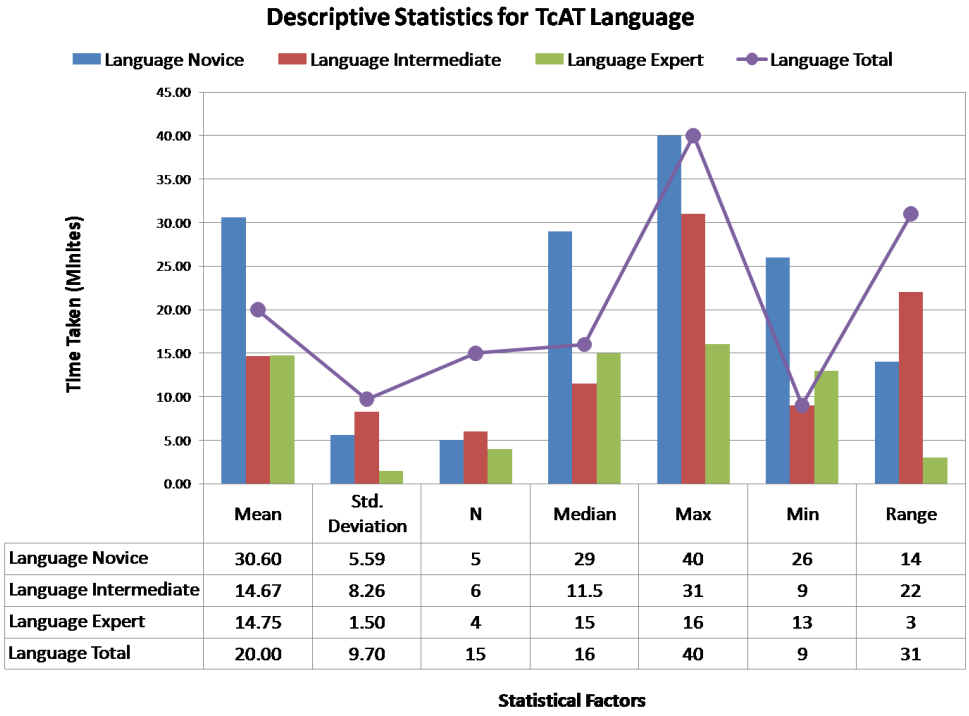
For TcAT GI, as shown in Figure 73, for average values, the novice group made the worst and expert group the best performance. The variation in values by the expert group is the slightest and by the intermediate is the most. Intermediate 4 took longer time than any novice subject. The maximum value of this method is by intermediate 4 who took an extremely long time. The maximum value by the expert group is less than the minimum by the novice group.



	Mean	Std. Deviation	N	Median	Max	Min	Range
TcAT Novice	33.00	3.39	5	34	37	29	8
TcAT Intermediate	23.33	8.38	6	22.5	38	15	23
TcAT Expert	16.00	2.45	4	15.5	19	14	5
TcAT Total	24.60	8.76	15	24	38	14	24

Figure 73: Descriptive Statistics for TcAT GI

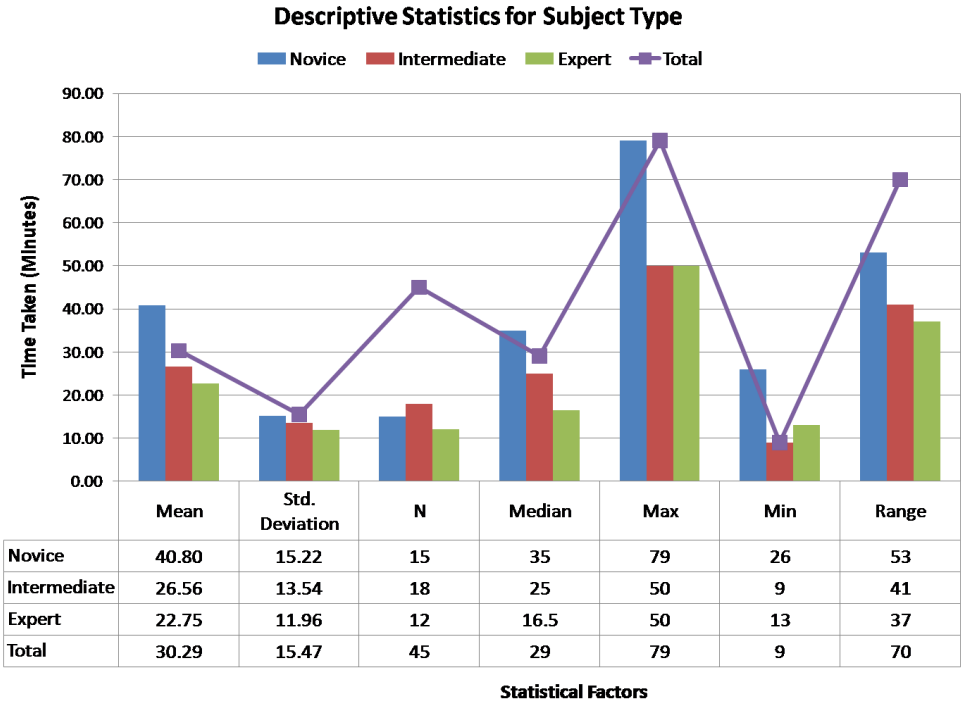
For TcAT language, as shown in Figure 74, for average values, the intermediate group made the best performance and the novice group made the worst. The variation in values by the expert group is the slightest and by the intermediate group is the most. Intermediate 4's poor performance made a major contribution to the greatest standard deviation and range values by the intermediate group. The maximum value of the expert group is less than the minimum of the novice group. Even with intermediate 4's poor performance, the intermediate group made the best performance.



	Mean	Std. Deviation	N	Median	Max	Min	Range
Language Novice	30.60	5.59	5	29	40	26	14
Language Intermediate	14.67	8.26	6	11.5	31	9	22
Language Expert	14.75	1.50	4	15	16	13	3
Language Total	20.00	9.70	15	16	40	9	31

Figure 74: Descriptive Statistics for TcAT Language

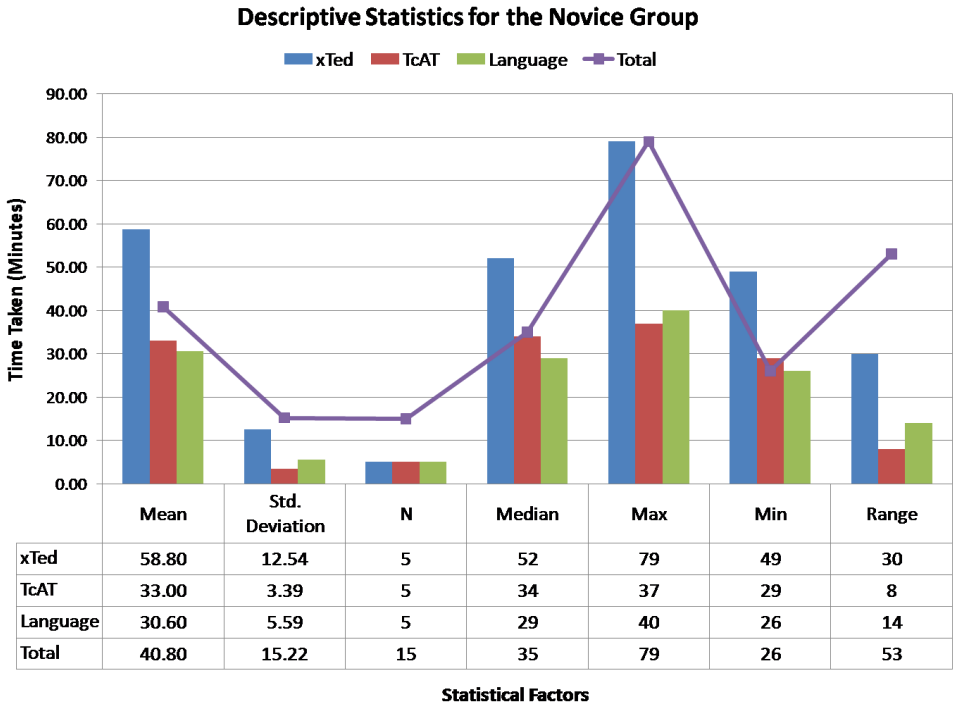
Figure 75 shows more detailed statistical information for subject type by using xTed, TcAT GI and TcAT language. On average, the novice group provides the worst performance and the expert group provides the best performance. Among the novice group, novice 2 showed the worst performance (79 minutes) with xTed and novice 2 showed the best performance (26 minutes) with TcAT language. Among the intermediate group, intermediate 5 showed the worst (50 minutes) with xTed and intermediate 6 showed the best performance (9 minutes) with TcAT language. Among the expert group, expert 3 made the worst (50 minutes) with xTed and expert 2 made the best performance (13 minutes) with TcAT language.



	Mean	Std. Deviation	N	Median	Max	Min	Range
Novice	40.80	15.22	15	35	79	26	53
Intermediate	26.56	13.54	18	25	50	9	41
Expert	22.75	11.96	12	16.5	50	13	37
Total	30.29	15.47	45	29	79	9	70

Figure 75: Descriptive Statistics for Subject Type

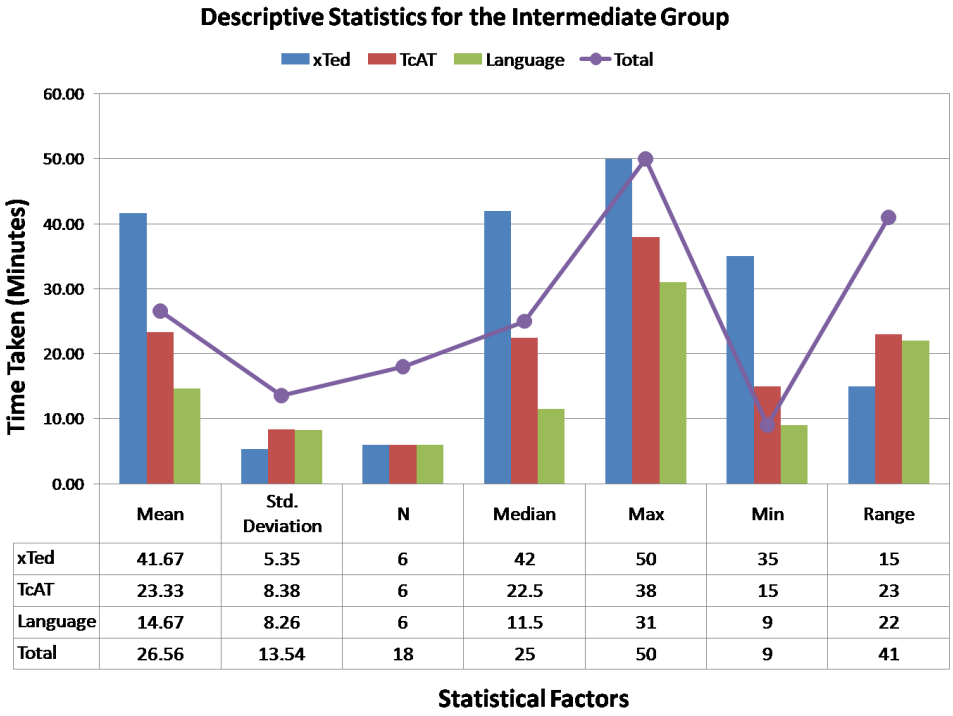
Figure 76, 77, and 78 show descriptive statistical information for each subject type by each method. For the novice group, as shown in Figure 76, for average values, with xTed made the worst and with TcAT language made the best performance. The variation among values and the range with TcAT GI are the slightest and with xTed are the most. Every novice subject took a longer time with xTed than the other two methods. Except for one subject (novice 1), every novice took a longer time with TcAT with GI than with textual language. Among the novice group, novice 2 showed the worst performance with xTed, and novice 1 made the worst performance with TcAT GI and textual language.



	Mean	Std. Deviation	N	Median	Max	Min	Range
xTed	58.80	12.54	5	52	79	49	30
TcAT	33.00	3.39	5	34	37	29	8
Language	30.60	5.59	5	29	40	26	14
Total	40.80	15.22	15	35	79	26	53

Figure 76: Descriptive Statistics for the Novice Group

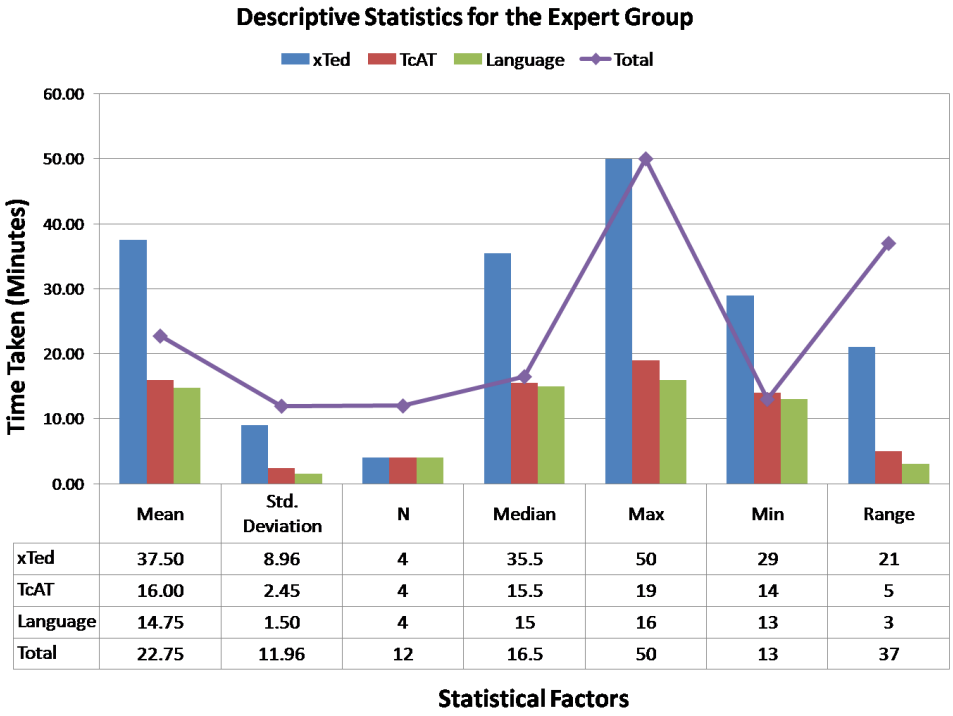
For the intermediate group, as shown in Figure 77, for average values, with xTed made the worst and with language made the best performance. The variation among values and range with xTed are the slightest and with TcAT GI and with language are similar values as the most. For every intermediate subject, using xTed was the worst and, using textual language was the best performance. Among intermediate group, intermediate 5 gave the worst performance with xTed, and intermediate 4 gave the worst performance with TcAT GI and textual language. Intermediate 4's TcAT GI value (38 minutes) is worse than xTed values of intermediate 1 (35 minutes) and intermediate 2 (37 minutes). The worst value with textual language is better than all performance with xTed.



	Mean	Std. Deviation	N	Median	Max	Min	Range
xTed	41.67	5.35	6	42	50	35	15
TcAT	23.33	8.38	6	22.5	38	15	23
Language	14.67	8.26	6	11.5	31	9	22
Total	26.56	13.54	18	25	50	9	41

Figure 77: Descriptive Statistics for the Intermediate Group

For the expert group, as shown in Figure 78, for average values, the performance order is textual language, TcAT GI, and xTed. The variation among values and the range with xTed are the most and with TcAT GI and with language are similar values as the slightest. The expert group with TcAT GI and with language showed the similar values for each statistical factor. Every expert subject gave the best performance with textual language. The worst values with TcAT GI and textual language are better than the best value with xTed.



	Mean	Std. Deviation	N	Median	Max	Min	Range
xTed	37.50	8.96	4	35.5	50	29	21
TcAT	16.00	2.45	4	15.5	19	14	5
Language	14.75	1.50	4	15	16	13	3
Total	22.75	11.96	12	16.5	50	13	37

Figure 78: Descriptive Statistics for the Expert Group

As presented in Figure 79, on average the novice group with xTed provided the overall worst performance (58.80 minutes) and the expert group with HTML provided the second worst performance. The intermediate group with textual languge provided the overall best performance (14.67 minutes). Table 12 shows the best and worst performance group for each authoring method. For total methods, the expert group provided the best and the novice group provided the worst performance.

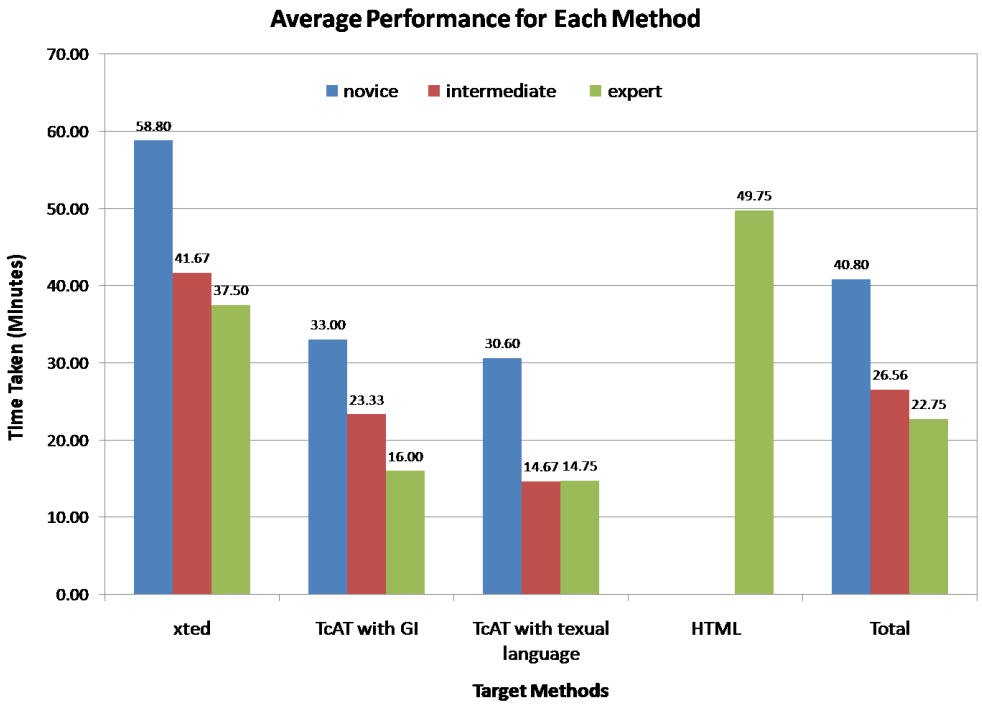


Figure 79: Average Performance for Each Method by Each Subject Type

Table 12 Best and Worst Subject Group for Authoring Method

Method	Best	Worst
xted	Expert	Novice
TcAT GI	Expert	Novice
TcAT Language	Intermediate	Novice
Total	Expert	Novice

Figure 80 presents the performance improvements from xTed to TcAT with GI, xTed to textual language, and from TcAT GI and textual language by each subject. Intermediate 5 provided 80% improvement from xTed to TcAT language, which is the best overall. As an extraordinary case, novice 1 showed -8.1% improvement from TcAT GI to language as the worst. Novice 1 and intermediate 4 showed poor improvements at

every comparison. Among the novice group, every novice except novice 1 and novice 2 showed similar improvements. Novice 1 showed poor improvements and novice 2 good improvements at every comparison. Among the intermediate group, every intermediate subject except intermediate 4 provided good improvement from xTed to TcAT language. Every intermediate subject except intermediate 1 and 4 provided good improvement from xTed to TcAT GI. Every intermediate subjects except intermediate 4 showed good improvements from TcAT GI to language. Among the expert group, every expert except expert 4 provided good improvement from xTed to TcAT GI and from xTed to TcAT language. Every expert showed poor improvements from TcAT GI to TcAT language.

Figure 81 presents the summary statistics for improvement data. For mean and median values, performance improvement from xTed to TcAT textual language showed the best and from TcAT GI to TcAT textual language showed the worst improvement. Improvement from TcAT GI to textual language showed the biggest difference from maximum value to minimum value since minimum value is extremely small.

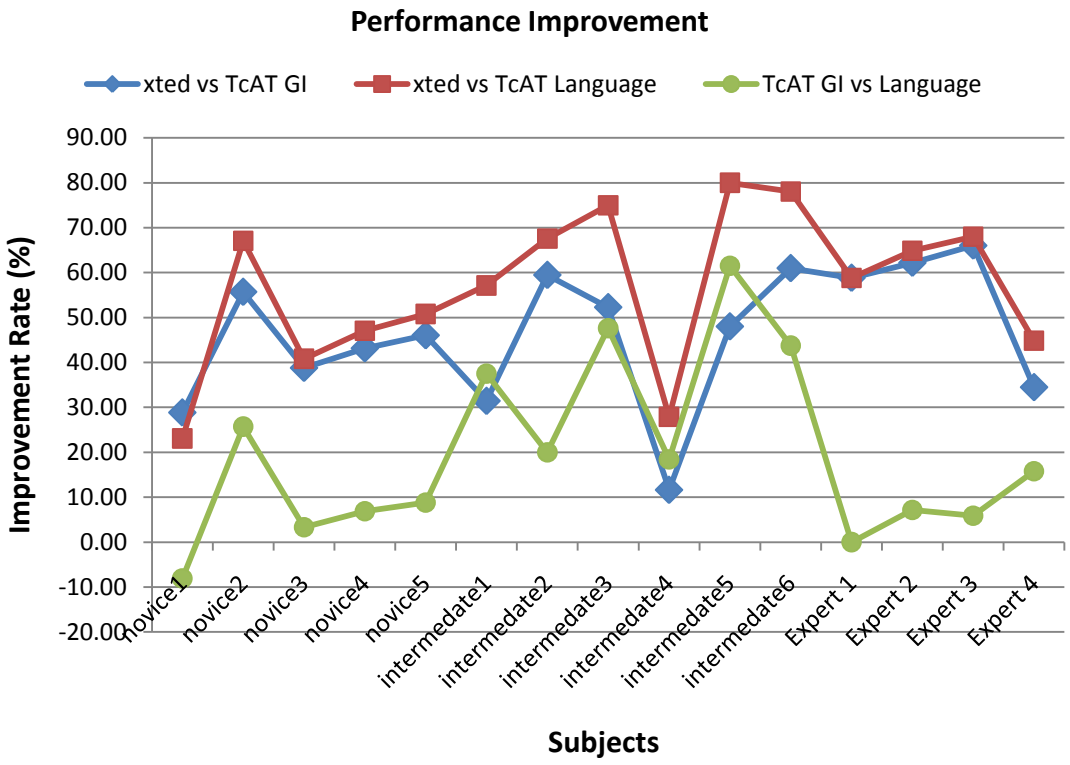
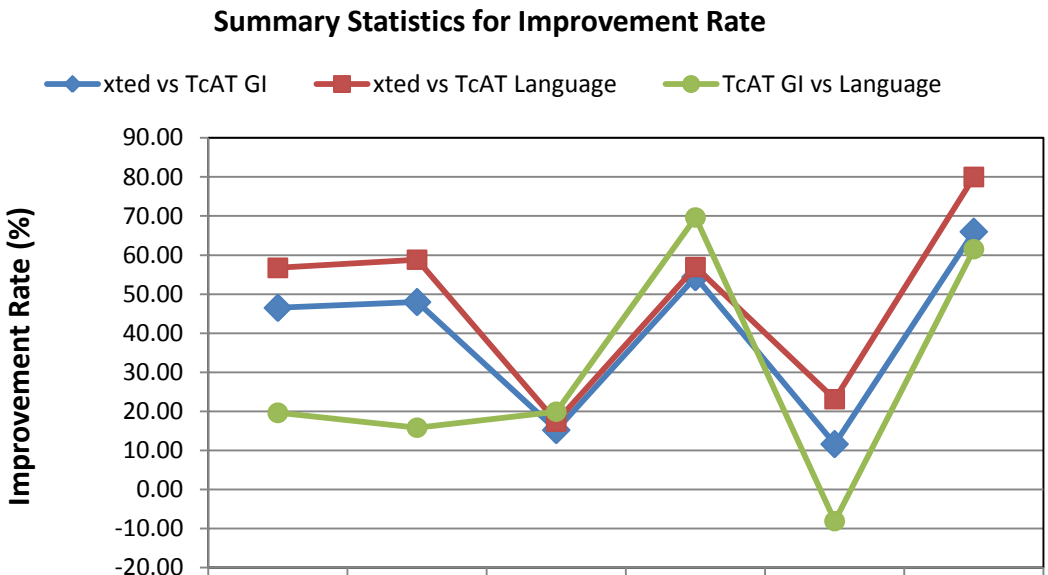


Figure 80: Performance Improvement

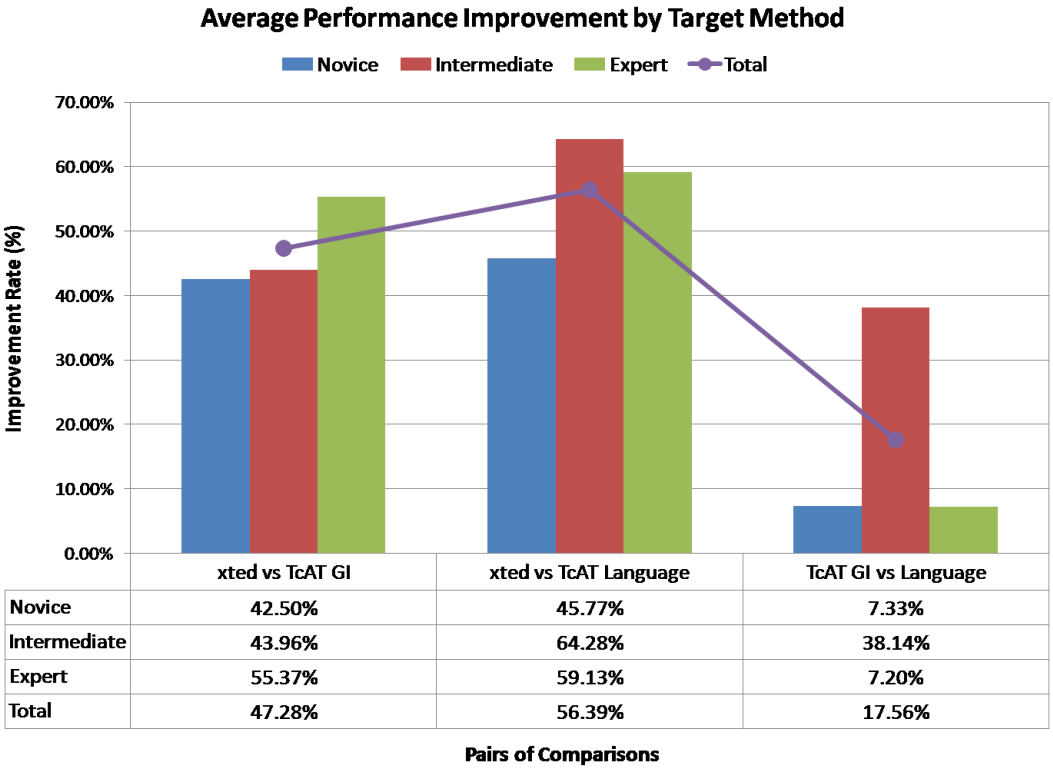


	Mean	Median	Standard Deviation	Range	Minimum	Maximum
xted vs TcAT GI	46.51	48	15.23	54.37	11.63	66
xted vs TcAT Language	56.73	58.82	17.40	56.92	23.08	80
TcAT GI vs Language	19.62	15.79	19.89	69.65	-8.11	61.54

Figure 81: Descriptive Statistics for Improvement Rate

As presented in Figure 82, which shows average improvement by each subject type, improvement from xTed to TcAT textual language by the intermediate group showed the best improvement overall. In case of xTed vs. TcAT GI, the expert group provided the best improvement among other subject type and the novice and the intermediate group showed the similar improvement. At comparison from xTed to TcAT textual language and from TcAT GI to TcAT textual language, the intermediate group showed the best performance improvement. In case of from xTed to TcAT language, the novice group showed a little bit lower improvement than the expert and intermediate

group. In case TcAT GI vs. language, the novice group and the expert group showed similar poor improvement. The intermediate group showed higher improvement.

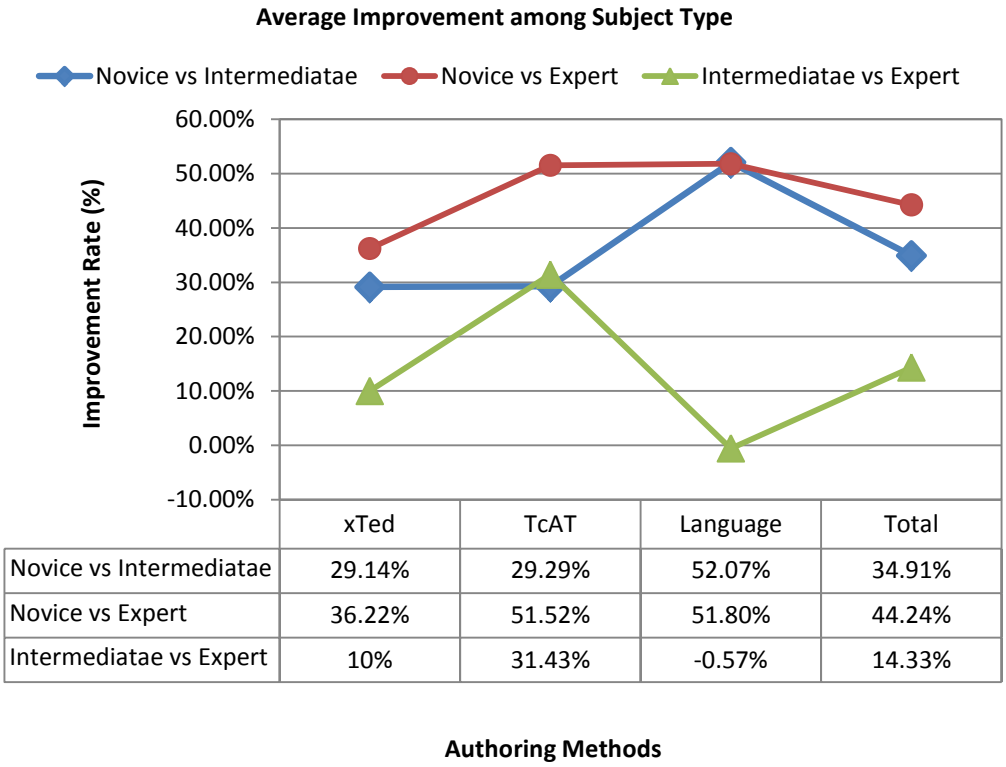


	xted vs TcAT GI	xted vs TcAT Language	TcAT GI vs Language
Novice	42.50%	45.77%	7.33%
Intermediate	43.96%	64.28%	38.14%
Expert	55.37%	59.13%	7.20%
Total	47.28%	56.39%	17.56%

Figure 82: Average Performance Improvement among Authoring Method

Figure 83 presents average performance improvements among subject types; from the novice group to the intermediate, from the novice to the expert, and the intermediate to the expert group. For total methods, the average improvement from the novice to the expert group showed the best and from the intermediate to the expert showed the worst performance. Using TcAT language, improvement from the novice to the intermediate group showed the best and from the intermediate to the expert showed the worst improvement. Improvement from the novice to the intermediate showed the

similar improvement using xTed and TcAT GI. This improvement showed high improvement using language. Improvement from the novice to the expert showed similar improvement using TcAT GI and TcAT language. Using xTed, this improvement showed lower improvement. From the intermediate to the expert showed good improvement using TcAT GI. Unpredictably, using language, the intermediate group showed the better performance than the expert group.



	xTed	TcAT	Language	Total
Novice vs Intermediatae	29.14%	29.29%	52.07%	34.91%
Novice vs Expert	36.22%	51.52%	51.80%	44.24%
Intermediatae vs Expert	10%	31.43%	-0.57%	14.33%

Figure 83: Average Performance Improvement among Subject Type

Figure 84 presents performance improvements when comparing from HTML to xTed, from HTML to TcAT GI and from HTML to language by each expert subject and average value. On average, the performance is increased by 21.62% (xTed), 67.61%

(TcAT GI), and 69.88% (textual language). For each expert, improvements from HTML to TcAT GI and from HTML to language showed similar high improvements even though improvements from HTML to xTed were a little bit diverse.

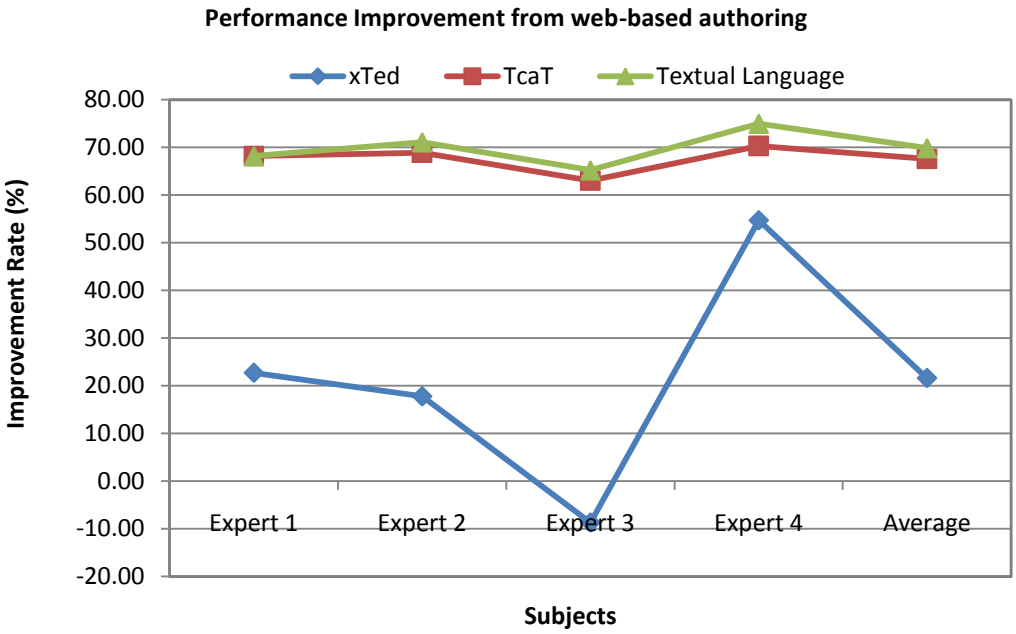


Figure 84: Performance Improvement from Web-based Authoring to Other Methods

For statistical analysis we took two way a 3×3 ANOVA test since the number of independent variables is two (subject type and authoring method) and the level of authoring method variable is three (xTed, TcAT, Textual language) and the level of subject type variable is three (novice, intermediate, expert). As dependent variable, we used time taken.

To use the ANOVA test, our data satisfies the one of the ANOVA assumptions, which is "homogeneity of variance", by failing to reject the null hypothesis (H_0) that is

there are no difference in the variance of different groups by using Levene's test of equality of error variances since 0.110 > 0.05. Table 13 shows the result of Levene's test.

Table 13: Result of Levene's Test

Levene's Test of Equality of Error Variances[a]

Dependent Variable:Time

F	df1	df2	Sig.
1.795	8	36	.110

Tests the null hypothesis that the error variance of the dependent variable is equal across groups.

a. Design: Intercept + Method + SubjectType + Method * SubjectType

In the ANOVA test, the following three hypotheses are used.

- H_{M0} = M_1 (mean of method 1) = M_2 (mean of method 2)=M_3 (mean of method 3)
- H_{S0} = S_1 (mean of subject type 1) = S_2 (mean of subject type 2)=S_3 (mean of subject type 3)
- H_{MS0}: There is no interaction between method and subject

From the results of the test presented in table 14, we reject H_{M0} and H_{S0} since both $p < 0.05$. So we can say there is a statistically significant difference among authoring methods and among subject types with $p<0.05$. We fail to reject H_{MS0} ($p=0.652$); therefore there is no interaction between method and subject. Based on adjusted R^2 value (0.779), 78% of the variance of "Time" variable can be explained by the method and subject type variables.

Based on post hoc tests presented in table 15, we can say there is a statistical significance in mean difference between xTed and TcAT and between xTed and textual

language with $p<0.05$. But there is no statistically significant mean difference between TcAT and textual language ($p=-.207$).

Table 14: Result of Two Way 3×3 ANOVA

Tests of Between-Subjects Effects

Dependent Variable:Time

Source	Type III Sum of Squares	df	Mean Square	F	Sig.	Partial Eta Squared
Corrected Model	8623.494[a]	8	1077.937	20.405	.000	.819
Intercept	39497.892	1	39497.892	747.692	.000	.954
Method	5694.939	2	2847.470	53.902	.000	.750
SubjectType	2590.150	2	1295.075	24.516	.000	.577
Method * SubjectType	130.633	4	32.658	.618	.652	.064
Error	1901.750	36	52.826			
Total	51809.000	45				
Corrected Total	10525.244	44				

a. R Squared = .819 (Adjusted R Squared = .779)

Table 15: The Result of Post Hoc Test for Authoring Methods

Multiple Comparisons
Time
Tukey HSD

(I) Method	(J) Method	Mean Difference (I-J)	Std. Error	Sig.	95% Confidence Interval	
					Lower Bound	Upper Bound
1	2	21.6667*	2.65396	0	15.1796	28.1537
	3	26.2667*	2.65396	0	19.7796	32.7537
2	1	-21.6667*	2.65396	0	-28.1537	-15.1796
	3	4.6	2.65396	0.207	-1.8871	11.0871
3	1	-26.2667*	2.65396	0	-32.7537	-19.7796
	2	-4.6	2.65396	0.207	-11.0871	1.8871

Based on observed means.
The error term is Mean Square(Error) = 52.826.
*. The mean difference is significant at the .05 level.

Based on post hoc tests presented in table 16, among subject type we can say there is a statistical significance in mean difference between novice and intermediate and between novice and expert with $p<0.05$. But there is no statistically significant mean difference between intermediate and expert ($p=-.349$).

Table 16: The Result Post Hoc Test for Subject Types

Multiple Comparisons
Time
Tukey HSD

(I) Subject Type	(J) Subject Type	Mean Difference (I-J)	Std. Error	Sig.	95% Confidence Interval	
					Lower Bound	Upper Bound
1	2	14.2444*	2.54098	0	8.0335	20.4553
	3	18.0500*	2.81495	0	11.1694	24.9306
2	1	-14.2444*	2.54098	0	-20.4553	-8.0335
	3	3.8056	2.70869	0.349	-2.8153	10.4264
3	1	-18.0500*	2.81495	0	-24.9306	-11.1694
	2	-3.8056	2.70869	0.349	-10.4264	2.8153

Based on observed means.
The error term is Mean Square (Error) =52.826.
*. The mean difference is significant at the .05 level.

5.3.2 Result of Summary Questionnaire

Figure 85 presents selected questions from our summary questionnaire that required the subject to rate each statement/opinion on a scale of 1 to 5 to verify the usefulness of proposed features. The complete questionnaire can be found in Appendix D. Figures 86, 87, 88, 89, and 90 show the results of this questionnaire. Figure 91 provides average value for each group of subjects and overall average value.

	Strongly Disagree		Neutral		Strongly agree
1: The proposed tool (TcAT) is useful for authoring large and complex hypertext.	1	2	3	4	5
2: Library net is useful for creating hypertext documents.	1	2	3	4	5
3: Semantics of composed net is useful for Composing hypertext from the existing nets	1	2	3	4	5
4: Automatic Petri net layout of composed hypertext from semantics of composed net is acceptable.	1	2	3	4	5
5: With using metadata, TcAT is useful in making a library net (component fragment).	1	2	3	4	5
6: With using metadata, TcAT is useful in finding a relevant library by using meta data.	1	2	3	4	5
7: TcAT is useful for modifying the existing hypertext.	1	2	3	4	5
8: Authoring language is useful for creating a hypertext document.	1	2	3	4	5
9: Automatic Petri net layout from authoring language is acceptable.	1	2	3	4	5
10: In using the Petri-net based hypertext system such as caT, I found the automatic conversion of the HTML page to the Petri-net based hypertext page is helpful.	1	2	3	4	5
11: Automatic conversion HTML page to the Petri-net based hypertext page is acceptable.	1	2	3	4	5
12: The user interface of TcAT is easy and efficient.	1	2	3	4	5
25: If the tool is available to you, would you use it?	1	2	3	4	5
26: How would you rate your experience with TcAT?	1	2	3	4	5

Figure 85: Rate Summary Questionnaires

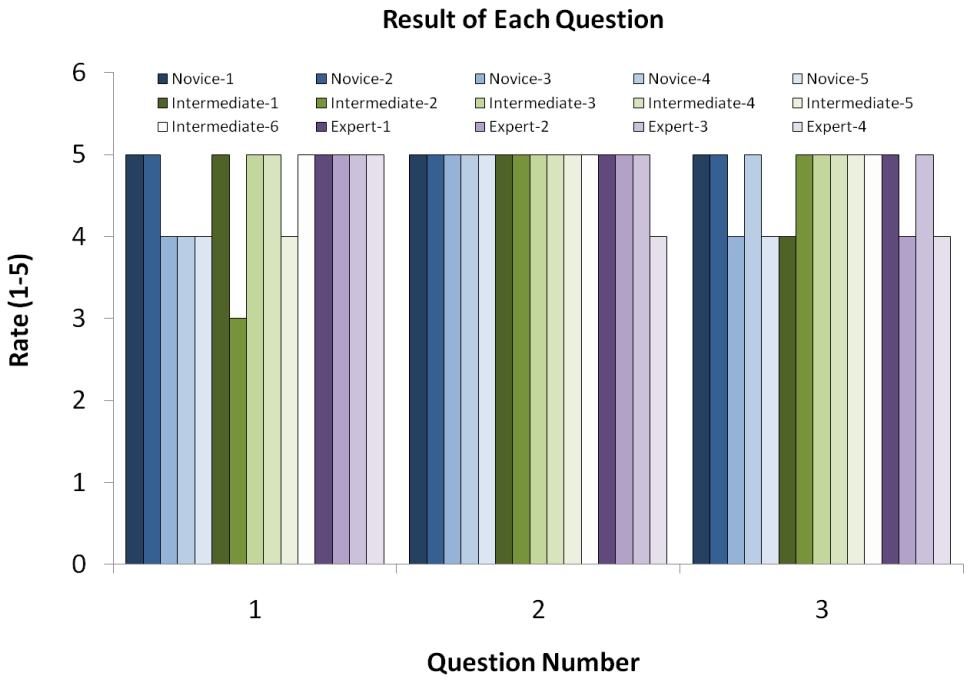


Figure 86: Results of Question 1 to 3

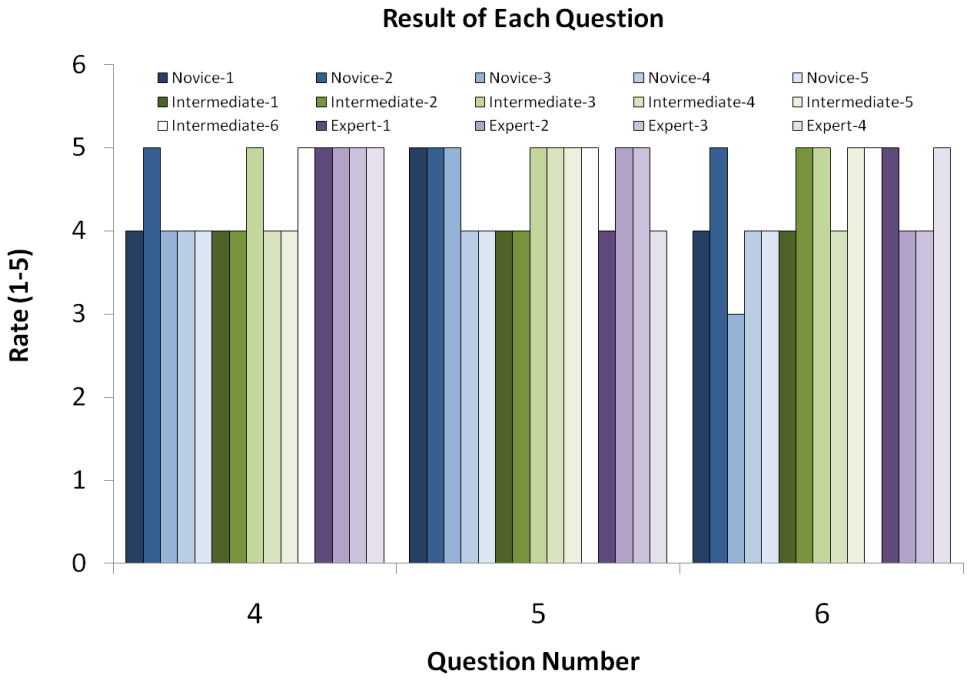


Figure 87: Results of Question 4 to 6

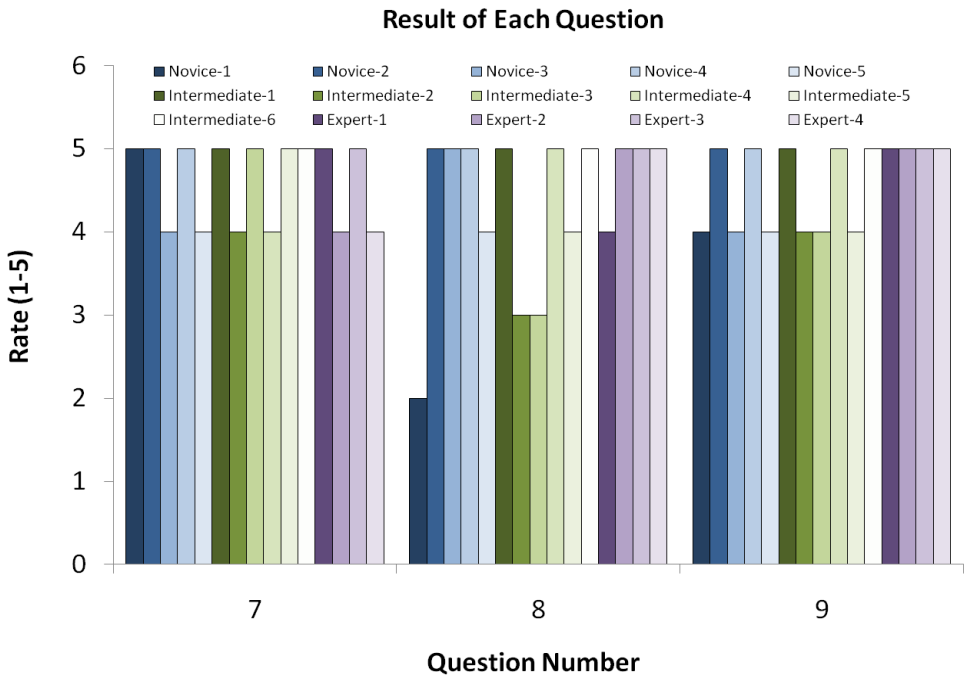


Figure 88: Results of Question 7 to 9

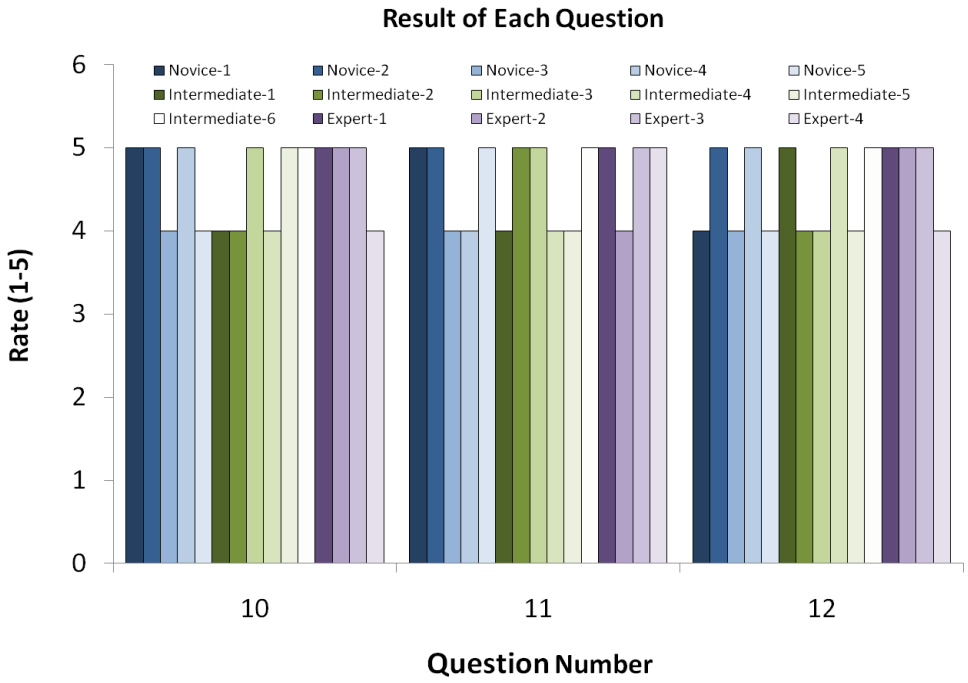


Figure 89: Results of Question 10 to 12

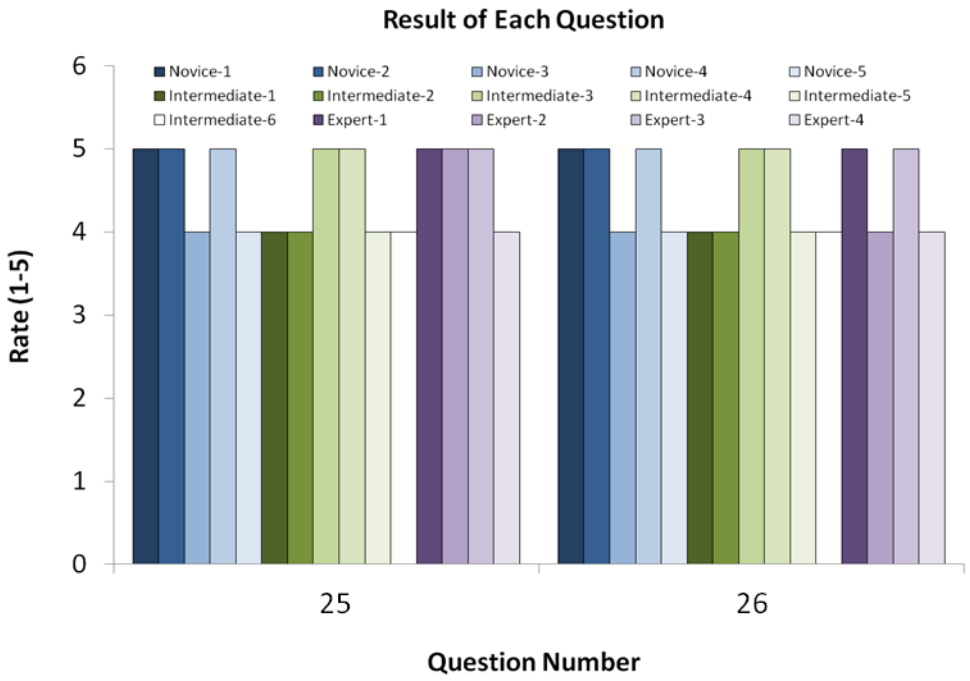


Figure 90: Results of Question 25 and 26

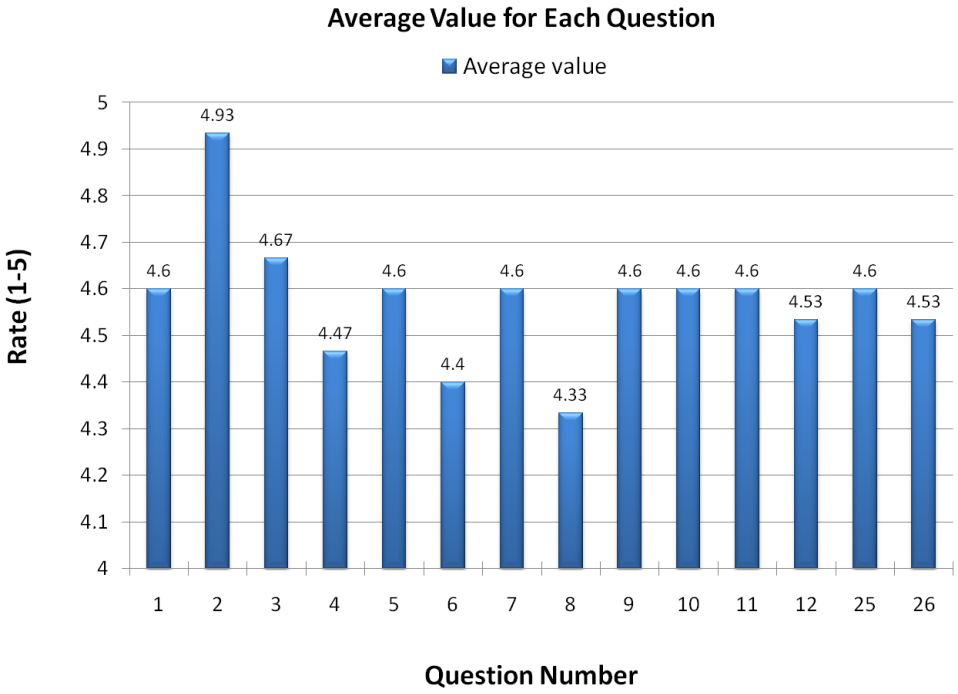


Figure 91: Average Value for Each Question

Except in a few cases, all proposed features are verified by subjects. In the case of question 8 asking the usefulness of authoring language, we recognize that novice 2 disagrees by giving rate 2 (see Figure 88). In the previous section, novice 2 shows an unusual performance by taking a longer time using the authoring language than using TcAT GI. We cannot say the authoring language is a useful authoring mechanism to novice 2. This is only case that the subject disagree the usefulness of proposed features.

5.4 Qualitative Data

In the free post-task summary questionnaire, we had twelve questions for each subject to write down her ideas and comments. Tables E.1 to E.12 in appendix E show each question and the answer for that question by each subject. Through analysis of the answers of these questions, we can recognize significant characteristics for Petri net-based authoring large and complex hypertext, which would be helpful information for future enhancement.

Figure 92 presents subjects' answers about the features of proposed tool that are helpful in creating large and complex hypertexts. Figure 93 presents features subjects would like to add to the prototype. Subjects' answers about the features of the prototype that make it difficult to create large and complex hypertexts are presented in Figure 94.

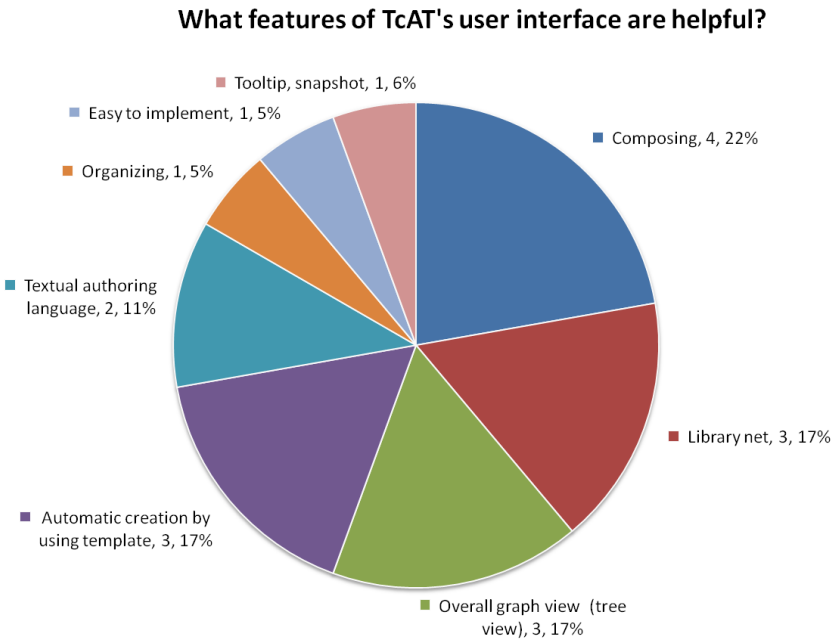


Figure 92: Features of TcAT Subjects Said Helpful

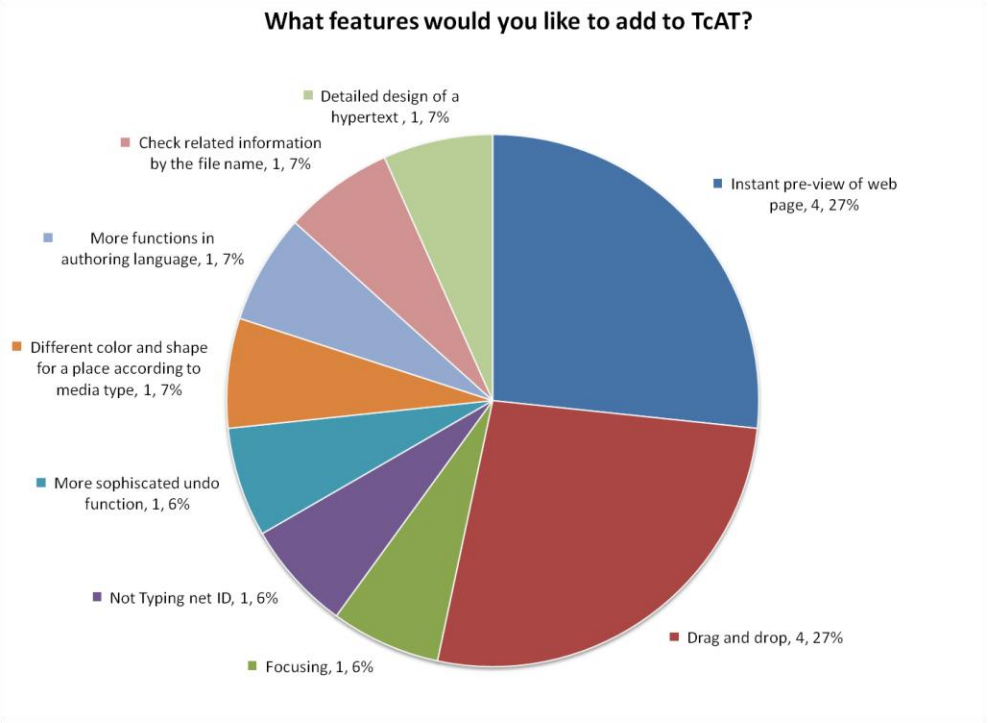


Figure 93: Features Subjects would like to Add to TcAT

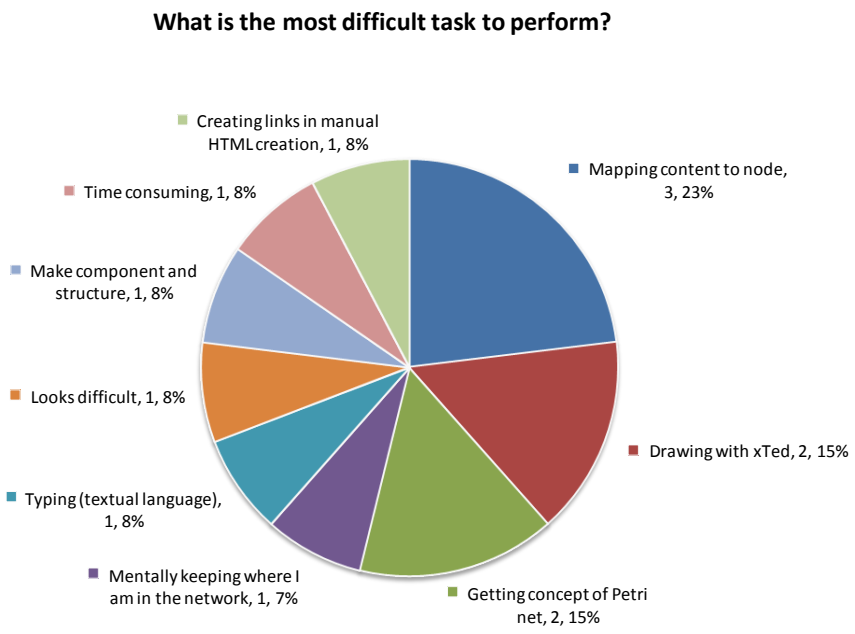


Figure 94: Most Difficult Task Subject to Perform

5.5 Patterns of Organizing Policy

When subjects organized the collections, they employed similar policies. As presented in Figure 95, five subjects sequentially organized the collections and four subjects categorized collections by using theme of paintings. Three subjects organized collections by periodically according to painting's date, two subjects organized by alphabetical order of the painting's title, and one subject organized using the year of the painting's date.

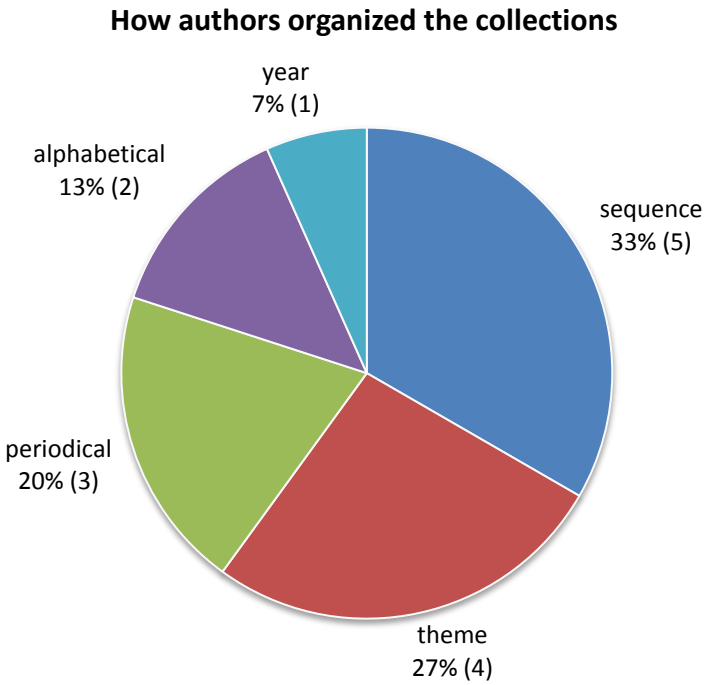


Figure 95: Patterns of Organizing the Collections

5.6 Discussion

This evaluation took around four to five hours to create a hypertext including tutoring in the tools and Petri net-based hypertext. It was very difficult for subjects to ask for other tasks such as search and modifying nets to evaluate due to time limit. For these tasks, subjects just had enough experience to answer the summary question. As future work, additional user evaluation for other tasks excluding creating hypertext would be helpful for analysis for this research.

We did not consider learning effects of repeated tasks for subjects even though learning effects are important factors for analyzing subject's behaviors. For future user evaluation we need more a sophisticated design.

CHAPTER VI

CONCLUSION AND FUTURE WORK

This chapter summarizes for contributions of this research. Discussions and future work follow.

6.1 Conclusion

To provide easier and more efficient authoring environments for caT/Petri-net-based hypertext system, we have introduced component-based specification. Component Petri net (CN) and predefined component net (template) support reuse of net fragments for building large networks. Component net uses a Petri net algebra and net transformation mechanism. By using the semantics of component nets, this research provides advanced compositional mechanisms for composing Petri nets as building blocks. Automatic Petri net creation by using template and library net helps authors creating nets.

We have provided other advanced authoring mechanisms that authors can easily use to create, locate, edit and save components/templates. Component fragment/library fragment is characterized by using metadata. This component fragment can be retrieved in a fragment repository using metadata. We locate the retrieved component fragment, and the fragment can be modified and expanded according to new requirements.

To enhance authoring mechanism for large and complex nets, a textual authoring language has been developed. The ability to edit nets visually as well as textually enables authors to modify large structures quickly and a two-way translation of the edits

to the net structure ensures that both views reflect the current structure. In addition to the visual interface, our textual authoring language completes using the compositional operations of CNs. The position of the Petri net is automatically computed and the Petri net is automatically displayed in the graphical editor.

To provide more user-friendly and more sophisticated functions than Motif-based xTed, a new authoring tool called TcAT (Template-based caT Authoring Tool) has been developed. To be a system independent tool that executes on multiple platforms, TcAT is implemented using Java. TcAT provides the following features for enhanced authoring environments:

- Library net
- Compositional mechanism
- Template panel to display current templates
- Different levels of abstraction by implementing collapsed/extended view
- Tabbed panels to display subnets for collapsed view
- Grouping and selecting mechanisms
- Collapsible tree view navigation of the entire network
- Tool tips for each net and net element
- Search mechanism
- Net modification mechanism
- Integration of other applications
- Text panel
- Content panel

- Zooming mechanism

To verify the provided features are useful for authoring large and complex hypertext in a Petri net-based hypertext system, we performed an usability evaluation. For this, we measured effectiveness of our interface by using time taken for authoring. In our evaluation, subjects authored a hypertext using TcAT significantly faster than when they used xTed (42.5% improvement for novices, 43.96% for intermediate group, and 55.4% for experts). The performance improvement of TcAT users was slightly higher when using the textual authoring language (45.8% for novices, 64.3% for intermediate and 59.1% for experts). TcAT authors preferred the textual language and quickly adopted its use.

The important features of TcAT were verified through subjects' rating of each statement on summary questionnaires. Through interviews with subjects we discovered the significant characteristics for authoring large and complex hypertext. We can tweak the existing features and develop new ones based upon feedback from subjects.

6.2 Discussions and Future Work

Digital libraries, museum, and collections employ several well-recognized, templatable features, such as on-line and off-line help, galleries, creation and browsing of personalized collections, specialized or advanced services for paying patrons, and additional privileges for collection managers. TcAT will enable authors to create involved digital library infrastructures that support multiple advanced features using the Petri net-based formalism.

Over time, we will continue to analyze the networks created by authors to assess the effectiveness of new features. We expect that these will encourage authors to create larger networks with greater ease. We also expect to modify the existing features and develop new ones based upon feedback from authors. For example, we need to provide more intuitive user interface mechanisms to TcAT such as a drag and drop mechanism to map contents to elements of net.

In this research, we have introduced conversion of web pages to the caT specification to provide caT authors an easy way to generate contextual presentations from pre-existing information. We provide automatic as well as semi-automatic transformation mechanisms to extract content and link structure from these documents. Web page patterns that model page genres lay a foundation for defining automatic transformation mechanisms. Transformation rules enable the system as well as the users to optimize the net structure for readability and ease of management. Generated nets can be refined further for more personalized contextual presentation.

In case automatic conversion, we just provided a transformation rule for "personal web page" pattern. As our tools mature, we will focus on providing more transformation rules for other patterns of web pages such as online news sites, portals, and digital museums. A pattern recognition mechanism will be required to identity the pattern of web pages.

REFERENCES

Alexander, C., Ishikawa, S., and Silverstein, M. 1977. *A Pattern Language: Towns, Buildings, Construction*, Oxford University Press, Oxford, UK.

Battista, G. D., Eades, P., Tamassia, R., and Tollis, I. G. 1994. Algorithms for drawing graphs: an annotated bibliography. *Computational Geometry: Theory and Applications*, 4, 5, (Oct.), 235-282.

Berthelot, G. 1987. Transformations and decompositions of nets. *Advances in Petri Nets 1986, Part I in Petri Nets: Central Models and Their Properties*, Springer-Verlag, London, UK, 359-376.

Best, E., Fleischhack, H., Fraczak, W., Hopkins, R. P., Klaudel, H., and Pelz, E. 1995. A class of composable high level Petri nets. In *Proceeding of the 16th International Conference on Application and Theory of Petri Nets*, Turin, Italy, (June), 103-120.

Best, E., Devillers, R., and Koutny, M. 2001. *Petri Net Algebra*. Springer-Verlag, New York, NY.

Carpano, M. J. 1980. Automatic display of hierarchized graphs for computer aided decision analysis. *IEEE Transactions on Systems, Man, and Cybernetics*, 10, 11, 705-715.

Chen, Y., Tsia, W. T. and Chao, D. Y. 1993. Dependency analysis-a Petri-net-based technique for synthesizing large concurrent. *IEEE Transactions on Parallel and Distributed Systems*, 4, 4, (April), 414-426.

Christensen, S., and Petrucci, L. 1992. Towards a modular analysis of colored Petri nets. In *Proceeding of the 13th International Conference on Application and Theory of Petri Nets*, Sheffield, UK, (June), 113-133.

Christensen, S., and Petrucci, L. 2000. Modular analysis of Petri nets. *Computer Journal*, 43, 3, 224-242.

Cruz, I. F., and Tamassia, R. 1984 Graph Drawing Tutorial. http://graphdrawing.org/literature/gd-constraints.pdf

Eades, P., and Sugiyama, K. 1990. How to draw a directed graph. *Journal of Information Processing*, 13, 4, 424-437.

Ellis, C. A., Keddara, K., and Rozenberg, G. 1995. Dynamic change within workflow systems. In *Proceedings of the Conference on Organizational Computing Systems*, Milpitas, CA (Aug.), 10-21.

Fleischhack, H., and Grahlmann, B. 1998. A compositional Petri net semantics for SDL. In *Proceeding of the 19th International Conference on Application and Theory of Petri Nets*, Lisbon, Portugal, (June) 144-164.

Furuta, R., and Stotts, P. D. 1989. Programmable browsing semantics in Trellis. In *Proceedings of the Second Annual ACM Conference on Hypertext*, Pittsburgh, PA, (Nov.), 27-42.

Furuta, R., and Na, J. C. 2002. Applying caT's programmable browsing semantics to specify World-Wide Web documents that reflect place, time, reader, and community.

In *Proceedings of the 2002 ACM Symposium on Document Engineering*, McLean, VA, (Nov.), 10-17.

Gamma, E., Helm, R., Johnson, R., and Vlissides, J. 1995. *Design Patterns: Elements of Reusable Object-oriented Software*, Addison-Wesley, Reading, MA.

Gansner, E. R., North, S. C., and Vo, K. P. 1988. DAG: a program that draws directed graphs. *Software Practice and Experience,* 18, 11, 1047-1062.

Hamadi, R., and Benatallah, B. 2003. A Petri net-based model for web service composition. In *Proceedings of the Fourteenth Australasian Database Conference*, Adelaide, Australia, (Feb.), 191-200.

Horie, I., and Yamaguchi, K. 2004. Structural analysis for web documentation using the non-well-founded set. In *Proceedings of the 15th ACM Conference on Hypertext and Hypermedia*, Santa Cruz, CA, 42-43.

Jensen, K. 1992. *Coloured Petri Nets - Basic Concepts, Analysis Methods and Practical Use, vol 1*. Springer-Verlag, New York, NY.

Karadkar, U., Na, J-C. and Furuta, R. 2002. Employing Smart Browsers to Support Flexible Information Presentation in Petri net-based Digital Libraries. In *Proceedings of European Conference on Digital Libraries* (ECDL 2002), Rome, (Sept.), Springer-Verlag LNCS 2458, 324-337

Karadkar, U., Furuta, R., Ustun, S., Park, Y., Na, J-C., Gupta, V., Ciftci, T. and Park, Y. 2004. Display-agnostic Hypermedia. In *Proceedings of the 15th ACM Conference on Hypertext and Hypermedia,* Santa Cruz, CA, 58-67.

Koh, I., and Dicesare, F. 1991. Modular transformation methods for generalized Petri nets and their application to automated manufacturing systems. *IEEE Transactions Systems, Man and Cybernetics,* 21, 6. 1512-1521

Lakos, C., and Keen, C. 1991. Modeling layered protocols in LOOPN. In *Proceedings of Fourth International Workshop on Petri Nets and Performance Models*, Melbourne, Australia, 106-115

Lakos, C., and Keen, C. 1994. LOOPN++: new language for object-oriented Petri nets. In *Proceedings of European Simulation and Modeling Conference (ESM'94)*, Barcelona, Spain, (June) 369-374.

Lakos, C. 1995. From coloured Petri nets to object Petri nets. In *Proceeding of the 16th International Conference on Application and Theory of Petri Nets*, Turin, Italy, (June), 278-297.

Lee, K. H., Favrel, J., and Baptiste, P. 1987. Generalized Petri net reduction method. *IEEE Transactions on Systems Man and Cybernetics*, 17, 2, 297-303.

Lilius, J. 2001. OB(PN)^2: an object based Petri net programming notation. *Concurrent Object-Oriented Programming and Petri Nets: Advances in Petri Nets (Lecture Notes in Computer Science),* Springer-Verlag,New York, NY, 247-275.

Meseguer, J., and Montanari, U. 1990. Petri nets are monoids. *Information and Computation*, 88, 2, (Oct.), 105-155.

Messinger, E. B. 1988. Automatic layout of large directed graphs. Tech. Rep. 88-07-08, Dept. of Computer Science, Univ. of Washington, Seattle, WA.

Na, J. C. 2001. Context-aware hypermedia in a dynamically changing environment, supported by a high-level Petri net. Ph.D. Dissertation, Texas A&M University, College Station, TX.

Naedele, M. and Janneck, J. W. 1998. Design patterns in Petri net system modeling, In *Proceeding of the 4th International Conference on Engineering of Complex Computer Systems (ICECCS '98)*, Monterey, CA, (August), 47-54.

Padberg, J. 1999. Categorical approach to horizontal structuring and refinement of high-level replacement systems. *Applied Categorical Structures*, 7, 4, (Dec.). 371-403

Pomello, L. 1987. Observing net behaviour. *Concurrency and Nets: Advances in Petri Nets*, Springer-Verlag, New York, NY, 403-422.

Reisig, W. 1991. Petri nets and algebraic specifications. *Theoretical Computer Science*, 80, 1-34.

Rowe, L. A., Davis, M., Messinger, E., Meyer, C., Spirakis, C., and Tuan, A. 1987. A browser for directed graphs. *Software Practice and Experience*, 17, 1, 61-76.

Schmidt, D. C. 1995. Using design patterns to develop reusable object-oriented communication software. *Communications of the ACM*, 38, 10, (Oct.), 65-74.

Stehno, C. 2002. System specification and verification using high level concepts. In *Proceedings of the 9th International SPIN Workshop on Model Checking of Software*, Springer-Verlag, London, UK, 246-249.

Stotts, P. D., and Furuta, R. 1989. Petri-net-based hypertext: document structure with browsing semantics. *ACM Transactions on Information Systems* (TOIS), 7, 1, (Jan.), 3-29

Stotts, P. D., and Furuta, R. 1990. Hierarchy, composition, scripting languages, and translators for structured hypertext. Tech. Rep. CS-TR-2472 and UMIACS-TR-90-68, Department of Computer Science and Institute for Advanced Computer Studies, University of Maryland, College Park, MD.

Sugiyama, K., Tagawa, S., and Toda, M. 1981. Methods for visual understanding of hierarchical system structures. *IEEE Transactions on Systems, Man and Cybernetics*, 11, 2, 109-125.

Suzuki, I., and Murata, T. 1983. A method for stepwise refinement and abstraction of Petri nets. *Journal of Computer and System Science*, 27, 1, 51-76.

Zhou, M. C., DiCesare, F., and Desrochers, A. A. 1989. A top-down modular approach to synthesis of Petri net models for manufacturing systems. In *Proceedings of the IEEE Robotics and Automation Conference*, Scottsdale, AZ, (May), 534-539,

APPENDIX A

IRB APPLICATION

Revised December 15, 2003

Texas A&M University
IRB Application
Protocol for Human Subjects in Research
Part I: Summary Cover Sheet

IRB # 2005-0559
(Internal use only)

If Requesting Exempt Status, Check Here ✓ (Exemp from Full Board Review)
Please check or provide details on the following information (enter N/A if not applicable)
New submission ✓ Re-submission ____ (IF protocol was disapproved)
Principal Investigator Name Yungah Park Faculty ____ Staff ____ Graduate Student ✓ Undergraduate Student ____
Department CPSC Mail Stop 3112 Phone (979)845-2572
Email pyungah@cs.tamu.edu Fax (979)847-8578
Is this study part of a Thesis or Dissertation? Yes ✓ No ____ If Yes, do you have Committee Approval? Yes ____ No ____

Co-Principal Investigator Name ____ Faculty ____ Staff ____ Graduate Student ____ Undergraduate Student ____
Department ____ Mail Stop ____ Phone ____
Email ____ Fax ____

Received
Research Compliance
OCT 31 2005
IRB

Graduate Committee Chair/Faculty Advisor Name (if student) Dr. Richard Furuta
Department CPSC Mail Stop 3112 Phone (979)845-3839
Email furuta@cs.tamu.edu Fax (979)847-8578

Project Title Usability evaluation for authoring large and complex hypertext with reusable components

Funding Agency N/A
Funding Administrator: RF ____ TAES ____ TEES ____ TAMU ____ TTI ____
Funding Status: Funded ____ Not Funded ✓ Pending ____ **(Please attach a copy of Grant Proposal)**
Funding Amount ____

Risk Management Matrix

		Probability That Something Will Go Wrong			
		A Likely to occur immediat ely or in a short period of time, expected to occur frequently	**B** Probably will occur in time	**C** May occur in time	**D** Unlikely to occur
Seriousness of Risk	**I** May result in death				3
	II May couse severe injury, major damage or loss, and/or result in negative publicity for the participant involved			3	2
	III Participation presents a minimal threat to safety, health and well-being of participants		3	2	
	IV No more than minimal risk	3	2		

Red Zone - 4 thru 5 Yellow Zone - 2 thru 3 Green Zone - 1

If your protocol falls in the **Yellow** or **Red Zone** please call (979) 458-3624 for further instructions.

Seriousnes of Risk IV **Probability That Something Will Go Wrong** D
These three fields must be answer

Activity	Associated Risks	Method to Manage
User Survey	Only mininal Risk	N/A
User Evaluation	Only mininal Risk	N/A

Page ____ of ____

Email irb@tamu.edu or call (979) 458-4067 with any questions regarding this form.

Revised December 15, 2003

Objective Estimate of Risk to Subject: None ✓ Low ____Moderate ____ High ____

Will Existing Documents Be Used? Yes ____ No ✓ Will Existing Specimens Be Used? Yes ____ No ✓

Research Methodology: Qualitative____Quantitative____ Both ✓

Gender of Subjects: Male ____ Female ____ Both ✓ Estimated Age of Subjects 18-60 Total Participants (est.) 20-80

Location of Research: Center for the study of Digital Library, HRBB, Texas A&M university

Subjects Recruited From:
____Psychology Subject Pool
____Other Subject Pool _____
✓ Other TAMU Students
✓ Community
____Women/Fetuses
____Children
____Treatment Centers
____Hospitals
____Prisoners
____Schools
____Others _____

Recruitment Method:
✓ Direct person-to-person contact
____Telephone solicitation (attach script)
____Newspaper Advertising (attach ad copy)
____Posted Notices (attach copy)
____Letter (attach copy)
____Other (describe)_____

Compensation for Subjects Yes____ No ✓ (If Yes, attach regular payment schedule)
Deception Used Yes____ No ✓ (If Yes, attach debriefing form)
Research/Course Credit for Subjects Yes____ No ✓

Invasive or Sensitive Procedures: Yes____ No ✓
____Blood Samples
____Urine Samples
____Physical Measurements (electrodes, etc.)
____Stress Exercise
____Review of Medical/Psychological Records
____rDNA
____Other (specify)_____

Sensitive Subject Matter: Yes____ No ✓
____Abortion
____AIDS/HIV
____Alcohol
____Body composition
____Criminal activity
____Depression
____ Drugs
____Other (specify)_____
____ Learning Disability
____ Psychological inventory
____ Sex
____ Suicide

Use of Video or Audio Taping Audio
If yes, answer the following:
Retained Yes ✓ No____
Length of time retained: Duration of Study
Destroy/Erase Yes ✓ No____
Other _____
Use specified in consent form Yes ✓ No____

Provisions for Confidentiality/Anonymity
____Replies Coded
____Secure Storage
✓ Anonymous Response OR
____Confidential
(Cannot be both anonymous and confidential)

Requesting waiver of signature on consent form. Yes ____ No ✓ If Yes, Attach justification for waiver request. Criteria for waiver request can be found in the Federal Regulation section 45 CFR 46.116 and 46.117 at the following web address: http://ohrp.osophs.dhhs.gov/humansubjects/guidance/45cfr46.htm#46.116
Location where consent forms will be filed: Center for the study of Digital Library, HRBB, Texas A&M university
(Consent forms must be kept on file for 3 years after the completion of the project. It is best to keep the forms in a campus office in a locked filing cabinet. If you are requesting a waiver of signature on the consent form, this question does not apply to you.)

Do you have any relationship with any or all of the subjects, other than your investigator role? Yes____ No ✓
If yes, you must explain the relationship in the "**Selection of Subjects**" section and how you will avoid any type of coercion (doctor-patient, teacher-student, counselor-student, etc.)

Abstract: Please provide a brief statement, in lay terminology, outlining the purpose of this study. *(Why you are doing this research project, and what you propose to learn.)*

Context aware Trellis (caT), like the Trellis hypertext system, uses Petri-net-based specification. caT hypertext clients extend Trellis to respond to characteristics of reader's physical and contextual environments. One of the main problems of caT is that authoring of a large system is still complex. Even though caT provides a hierarchical authoring mechanism by using a hierarchical Petri net, it does not provide flexible reuse of specification of component fragments, which limits its usefulness in constructing a large Petri net. caT does not have mechanisms for selecting the relevant specification and modifying the selected specification according to new requirements. Besides, current caT does not provide sophisticated compositional mechanisms for using Petri nets as building blocks. The lack of such functions is especially problematic with a large collection of component fragments, and reduces the usability of caT. To solve these problems, a new prototype for authoring hypertext with reusable components is proposed.
This study will evaluate this proposed prototype. How effectively users can author large and complex hypertext with this prototype will be measured through this study.

Email irb@tamu.edu or call (979) 458-4067 with any questions regarding this form.

Revised December 15, 2003

REQUEST FOR EXEMPTION from full IRB review

Some research projects involving human subjects are exempt from full review by the IRB. See the attached sheet on research categories exempt from full IRB review. *(Sensitive topics and subjects such as children or minors, pregnant women and prisoners are not considered for exempt research).*

Basis for Exemption [Please refer to attached "Categories Exempt From Full IRB Review."]
(Do not check unless requesting an exemption from full IRB review.)

_____Established Educational Settings/Normal Educational Practices(a letter of approval from a school official must be obtained and submitted to the IRB before the study can be conducted)**(studies with children or minors are not exempt)**

_____ Use of educational anonymous tests (cognitive, diagnostic, aptitude, advancement; **attach copy).**

__✓__ Survey or interview procedures, **[unless** identifying subjects places them at legal or personal risk, and unless survey or procedures deal with sensitive matters of personal behavior]

_____ Observations of public behavior **[unless** identifying subjects places them at legal or personal risk, and unless observations deal with sensitive matters of personal behavior]

_____ Anonymous collection or study of existing documents, records, pathological or diagnostic specimens which are without any identifiers or codes.

_____ Evaluation of agencies and programs for administrative purposes where there was no deviation from standard practice.

_____ Taste and food quality evaluation and consumer acceptance studies.

The U.S. population is becoming increasingly culturally, linguistically, economically, and ethnically diverse. The research needs to make a concerted effort to ensure that research subjects reflect the population demographically, including these groups who have been traditionally under represented. However, it is recognized that the available pool of subjects may preclude having a balanced population. If you cannot use a diverse population in your research, you must justify this action in Part II, A, 1.

NOTE: The IRB makes the final decision whether or not a proposal is exempt from full IRB review.

Please check with the IRB Program Coordinator (979-458-4067). Exempt proposals require an original and two (1) copies of each instrument, i.e., Part A, Part B (with signatures), consent forms, research instrument, recruitment materials, etc. Full IRB review proposals require an original, with signatures, and 3 full copies, including research instrument, consent forms, recruitment materials, etc.

Email irb@tamu.edu or call (979) 458-4067 with any questions regarding this form.

Revised December 15, 2003

Part II:

Part A

I have read the Belmont Report, "Ethical Principles and Guidelines for the Protection of Human Subjects of Research" and subscribe to the principles it contains. In light of this Declaration, I present for the Board's consideration the following information, which will be explained to the subject about the proposed research.

Signature 

Principal Investigator Yung Ah Park

1. **Selection of Subjects**

   a. Source and number

   20 to 80 subjects will be recruited from Texas A&M students and the Bryan-College station community.

   b. Method of recruitment and selection

   Direct person-to-person contact

   c. Ages and gender

   Age: 18-60, Male and female subjects

   d. Compensation

   No compensation

Revised December 15, 2003

e. Location and duration of experiment

The study will be conducted in the Center for the Study of Digital Library, HRBB, Texas A&M University. It will take 4 hours to complete the experiment.

f. Specific steps to ensure confidentiality or anonymity of responses of results

A random identifier not related with subject's name and characteristics will be used for subjects. Only research team members can access the information and responses of the subjects. Every document related with this study will be stored in the Center for the Study of Digital library, HRBB, Texas A&M University. All information that can identify the subjects will be removed when results of this study are presented in public.

g. The investigator's relationship to subjects

No relationship

2. Purpose of study

Context aware Trellis (caT), like the Trellis hypertext system, uses Petri-net-based specification. caT hypertext clients extend Trellis to respond to characteristics of reader's physical and contextual environments. One of the main problems of caT is that authoring of a large system is still complex. Even though caT provides a hierarchical authoring mechanism by using a hierarchical Petri net, it does not provide flexible reuse of specification of component fragments, which limits its usefulness in constructing a large Petri net. caT does not have mechanisms for selecting the relevant specification and modifying the selected specification according to new requirements. Besides, current caT does not provide sophisticated compositional mechanisms for using Petri nets as building blocks. The lack of such functions is especially problematic with a large collection of component fragments, and reduces the usability of caT. To solve these problems, a new prototype for authoring hypertext with reusable components is proposed.

The purpose of this study is to evaluate this proposed prototype. How effectively users can author large and complex hypertext with this prototype will be mainly evaluated through this study. A comparative analysis with existing methods will be used to measure the achievement of this proposed system.

Revised December 15, 2003

3. Research procedures

1) Introduce this study to a subject - objectives and procedures will be explained to the subjects.
2) Request for the subject to sign the consent (mandatory) and Audio tape release (optional) forms
3) Assign the random identifier to the subject
4) Request for the subject to fill out a demographic questionnaire (attached)
5) Request for the subject to perform specific assigned tasks (attached) by the researcher
6) Request for the subject to answer the questions regarding each task after performing the task ->Free-form Q&A
7) Request for the subject to answer the summary questionnaire (attached) for this study.
8) Ask the subjects about reasons of their answers, responses and behaviors -> Free-form Q&A
The more detailed protocol is attached.

a. Physical/Behavioral Aspects

The experiment is usability evaluation of the newly developed (proposed) authoring tool by the researcher. Users will be asked to perform assigned tasks (attached) with previous authoring tool (xTed) and the proposed tool to compare two tools. The researcher will observe the behaviors and responses of the subjects.

b. Deception of Coersion

There will be no deception and coercion procedures in this study.

4. Risks and Benefits to Subjects

a. A description of any potential risks of discomforts to the subject.

The subjects will be not exposed to any additional risks and discomforts. Only minimal risk and discomforts to the participant will be expected.

b. A definition of benefits to the research subject or alternatives for participation in the study.

Any benefits will not be provided to the subjects.

c. Do not include broad benefits to society of potential research benefits to a group as a benefit to the subjects.

Revised December 15, 2003

Part B.

SIGNATURE ASSURANCE: ***(this should be the last page of the protocol application before attachments)***

Principal Investigator/Graduate Student Assurance Statement:

I understand Texas A & M University's policy concerning research involving human subjects and I agree:

1. To accept responsibility for the scientific and ethical conduct of this research study;
2. To obtain prior approval from the Institutional Review Board before amending or altering the research protocol or implementing changes in the approved consent form:
3. To immediately report to the IRB any serious adverse reactions and/or unanticipated effects on subjects which may occur as a result of this study;
4. To complete, on request by the IRB, the Continuation/Final Review Forms.

SIGNATURE: _____ DATE: 10/27/05

TYPED NAME: Yung Ah Park E-MAIL: pyungah@cs.tamu.edu

***Faculty/Research Advisor's Assurance Statement:**

I certify that I have read and agree with this proposal, that the PI has received adequate training to perform this research, and will receive adequate supervision while performing this research.

SIGNATURE: _____ DATE: 10/27/05

TYPED NAME: Richard Furuta E-MAIL: furuta@cs.tamu.edu

*** If the principal investigator is completing this project to meet the requirements of a Texas A & M University academic program, or is a student, both the student's faculty/research advisor and the departmental head should sign the Signature Assurance Sheet.**

****Department Head**

This is to certify that I have reviewed this research protocol and agree that the research activity is within the mission of the Department and appropriate for the responsibilities and assigned duties of the principal investigator.

SIGNATURE: Valerie E. Taylor DATE: 10/27/05

TYPED NAME: Valerie E. Taylor E-MAIL: taylor@cs.tamu.edu

****If the principal investigator is also the Department Head, the College Dean or equivalent should sign the Signature Assurance Sheet.**

Email irb@tamu.edu or call (979) 458-4067 with any questions regarding this form.

APPENDIX B

INFORMED CONSENT DOCUMENT

CONSENT FORM
Usability evaluation for authoring large and complex hypertext with reusable components

I have been asked to participate in a research study, which is usability evaluation of authoring prototype for large and complex hypertext with reusable components. Whether effectively users can author large and complex hypertext using this prototype will be mainly evaluated through this study. I was selected to be a possible participant because a random identifier not related with my name and characteristics will be used. A total of 20-80 people have been asked to participate in this study. The purpose of this study is to verify the proposed features are useful for authoring large and complex hypertext in Petri net-based hypertext system. Also, this study will be helpful to discover useful features for authoring hypertext.

If I agree to be in this study, I will be asked to fill out a demographic questionnaire, to perform specific assigned tasks by the researcher (The researcher will observe the my behaviors and responses when I perform these tasks), to answer the questions regarding each task after performing the task, to answer the summary questionnaire for this study, and to answer reasons of my answers, responses and behaviors (Free-form Q&A). Parts of my interaction during the study may be recorded using audio tapes. If I do not want to be audio taped, I can still participate in the study without any consequences. This study will only take 4 hours. I will not be exposed to any risks in the study. There will be no immediate and direct benefit to me. I will not be compensated for my participation.

This study is anonymous. All information that can identify the subjects will be removed when results of this study are presented in public. The records of this study will be kept private. No identifiers linking me to the study will be included in any sort of report that might be published. Research records will be stored securely and only Yung Ah Park and caT (context aware Trellis) research team members will have access to the records including audio tapes.

My decision whether of not to participate will not affect my current or future relations with Texas A&M University. If I decide to participate, I am free to refuse to answer any of the questions that may make me uncomfortable. I can withdraw at any time with out my relations with the university, job, benefits, etc., being affected. I can contact either Yung Ah Park, 344 H. R. Bright Building, Texas A&M University, College Station, TX 77843-3112, Phone: (979) 845-2572, Email: pyungah@cs.tamu.edu or Dr. Richard Furuta, 402C H. R. Bright Building, Texas A&M University, College Station, TX 77843-3112, Phone: (979) 845-3839 Email: furuta@cs.tamu.edu with any questions about this study.

This research study has been reviewed by the Institutional Review Board- Human Subjects in Research, Texas A&M University. For research-related problems or questions regarding subjects' rights, I can contact the institutional Review Board through Dr. Michael W. Buckley, Director of Research Compliance, Office of Vice President for Research at (979) 845-8585 (mwbuckley@tamu.edu). I have read the above information. I have asked questions and have received answers to my satisfaction. I have been given a copy of this consent document for my records. By signing this document, I consent to participate in the study.

Signature:_____Date: _____

Signature of Investigator:_____ Date: _____

APPENDIX C

PRE-TASK QUESTIONAIRE

Usability Evaluation for Authoring Large and Complex Hypertext with Reusable Components

Pre-Task Questionnaire

Alias Used:_____ Date: _____

Instructions:

- Please circle the letter/number that you think is the best answer for a given question.
- Please do not answer a question if it makes you uncomfortable or you would not like to answer it for any other reason.

Personal Information

1) Gender
   a) Male
   b) Female

2) Age group
   a) 18-25
   b) 26-35
   c) 36-45
   d) 45-60

3) Race/Ethnicity
   a) Caucasian
   b) Black
   c) Native American
   d) Hispanic
   e) Asian/Pacific Islander
   f) Multicultural
   g) Other _____
   h) Do not wish to disclose

4) Nationality

   _____

5) Academic background
   a) Engineering
   b) Pure Sciences
   c) Social Sciences
   d) Architecture
   e) Business
   f) Education
   g) Other (Please Specify)_____

Computers

6) How long have you used computers?
   a) Less than 6 months
   b) 6 months to a year
   c) a year to two years
   d) more than two years
   e) No experience

7) How often do you use a computer?
   a) Daily (almost everyday)
   b) Weekly (2 to 3 times a week)
   c) Monthly (2 to 3 times a month)
   d) Less than once a month

8) What type of computer do you use? (please circle all that apply)
   a) Apple Macintosh
   b) Windows XP/Windows NT
   c) Unix workstations (Sun, HP, Linux, SGI, etc.)
   d) Other (Please Specify)_____

Hypertext, Petri net and its Authoring Tools

9) How long have you authored (created) web pages ?
   a) Less than 6 months
   b) 6 months to a year
   c) a year to two years
   d) more than two years
   e) No experience

10) Have you known Petri net before evaluation?
   a) Yes b) No

11) If you know Petri net, where did you learn? _____

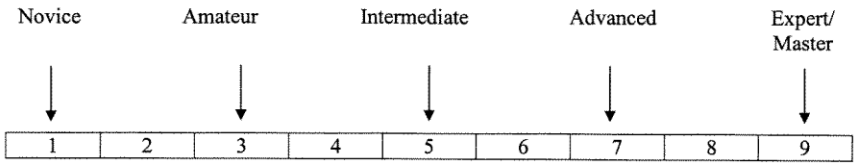
12) Have you ever drawn a Petri net?
   a) Yes b) No

13) If you have drawn a Petri net, what kinds of method did you used?
   a) By hand
   b) By using Authoring (Editor) Tool
   c) Both

14) If you have used Petri net tools, Please specify all

_____

15) How do you rate your knowledge about Petri net?



16) Have you created Hypertext by using Petri net Authoring (Editor) Tool?
a) Yes b) No

APPENDIX D

EVALUATION QUESTIONAIRE

Summary Questionnaire

Usability Evaluation for Authoring Large and Complex Hypertext with Reusable Components

Subject ID: _____

		strongly disagree		neutral		strongly agree
1	The proposed tool (TcaT) is useful for authoring large and complex hypertext.	1	2	3	4	5
2	Library net is useful for creating hypertext documents.	1	2	3	4	5
3	Semantics of component net is useful for composing a hypertext from the existing nets.	1	2	3	4	5
4	Automatic Petri net layout of composed hypertext from semantics of component net is acceptable.	1	2	3	4	5
5	With using metadata, TcaT is useful in making a library net (component fragment)	1	2	3	4	5
6	With using metadata, TcaT is useful in finding a relevant library by using meta data	1	2	3	4	5
7	TcaT is useful for modifying the existing hypertext.	1	2	3	4	5
8	Authoring language is useful for creating a hypertext document.	1	2	3	4	5
9	Automatic Petri net layout from authoring language is acceptable.	1	2	3	4	5
10	In using the Petri-net based hypertext system such as caT, I found the automatic conversion of the HTML page to the Petri-net based hypertext page is helpful.	1	2	3	4	5
11	Automatic conversion HTML page to the Petri-net based hypertext page is acceptable.	1	2	3	4	5
12	The user interface of TcaT is easy and efficient.	1	2	3	4	5
13	What features of TcaT's user interface are helpful ?					
14	What kinds of template nets are useful ?					
15	The feature I like most about TcaT compared to xTed?					
16	The feature I like least about TcaT compared to xTed?					
17	The feature I like least about xTed?					
18	The feature I like least about TcaT?					

19	What is the most difficult task to perform?					
20	What is the least difficult task to perform?					
21	What features of TcaT make it difficult to create hypertexts?					
22	What features of TcaT are most helpful in creating hypertexts?					
23	What features would you like to add to the TcaT?					
24	What features would you like to remove from TcaT?					
25	If the tool is available to you, would you use it?	1	2	3	4	5
26	How would you rate your experience with TcaT.	1	2	3	4	5

APPENDIX E

USER TESTING RESULT OF QUESTION 13 to 24

Table E.1: Answers for Question 13

Question 13	What features of TcAT's user interface are helpful?
Novice-1	Composing two sequences together
Novice-2	Composing, Library net, Tool tip, Snapshot, Net tree view
Novice-3	Library net
Novice-4	Composing
Novice-5	Library net
Intermediate-1	Creation of Petri net using templates
Intermediate-2	Text editor (authoring language)
Intermediate-3	Automatic creation by using template
Intermediate-4	Overall graph view and easy merge
Intermediate-5	Tree view since it make it easy to know whole structure
Intermediate-6	No answer
Expert-1	No answer
Expert-2	Net generated automatically, and easy to implement
Expert-3	Organizing
Expert-4	Text editor

Table E.2: Answers for Question 14

Question 14	What kinds of template net are useful?
Novice-1	Library nets
Novice-2	Library nets
Novice-3	Library net
Novice-4	Library net
Novice-5	Library net
Intermediate-1	Sequence and choice library net
Intermediate-2	Library net
Intermediate-3	Choice
Intermediate-4	Choice
Intermediate-5	No answer
Intermediate-6	No answer
Expert-1	Choice/parallel
Expert-2	No answer
Expert-3	No answer
Expert-4	No answer

Table E.3: Answers for Question 15

Question 15	The feature I like most about TcAT compare to xTed?
Novice-1	Composing many sequences together
Novice-2	Composing
Novice-3	Language
Novice-4	Library net
Novice-5	Building blocks
Intermediate-1	Template and authoring language
Intermediate-2	Modulization
Intermediate-3	Choice and parallel
Intermediate-4	Suggested structure and easy merge, combine
Intermediate-5	Easy to combine and search, automatic creation of component
Intermediate-6	No answer
Expert-1	Template
Expert-2	Easy to implement and Save time
Expert-3	Componentized
Expert-4	Net creation by using library

Table E.4: Answers for Question 16

Question 16	The feature I like least about TcAT compare to xTed?
Novice-1	Number of internode
Novice-2	Needs learning for TcAT to use
Novice-3	Hard to remember node ID
Novice-4	Language
Novice-5	Net number remembering
Intermediate-1	N/A
Intermediate-2	Difficult to find
Intermediate-3	Nothing
Intermediate-4	Nothing
Intermediate-5	Nothing
Intermediate-6	No answer
Expert-1	Nothing
Expert-2	No answer
Expert-3	No answer
Expert-4	Displaying null node

Table E.5: Answers for Question 17

Question 17	The feature I like least about xTed?
Novice-1	Drawing
Novice-2	Too much work and time spending (Simple hypertext is fine, but when it gets larger, hard to work)
Novice-3	Repeated drawing
Novice-4	Drawing is boring
Novice-5	Too much hand work
Intermediate-1	Not efficient to build Petri net one-by-one
Intermediate-2	User interface/ undo function
Intermediate-3	No answer
Intermediate-4	Structure and transition always create
Intermediate-5	Inefficient, difficult to handle when drawing is large
Intermediate-6	No answer
Expert-1	Need to design all components one by one
Expert-2	A tedious work
Expert-3	Tedious task
Expert-4	It takes time to create many nets

Table E.6: Answers for Question 18

Question 18	The feature I like least about TcAT?
Novice-1	Number of internode is confusing
Novice-2	It would be helpful to see by differentiating places. (Depending content type (text or media), provides different shape or color)
Novice-3	Need instant preview of web-page
Novice-4	Language
Novice-5	Net number remembering
Intermediate-1	None
Intermediate-2	Undo function
Intermediate-3	Provide different colors for nets when composing the nets
Intermediate-4	No drag and drop
Intermediate-5	Incorrect combine
Intermediate-6	No answer
Expert-1	Need to use bottom-up design method to fully use TcAT
Expert-2	Typing each node ID may be very tedious work in case of lots of node ID.
Expert-3	No answer
Expert-4	Hard to type net ID

Table E.7: Answers for Question 19

Question 19	What is the most difficult task to perform?
Novice-1	Drawing
Novice-2	Mapping
Novice-3	Define concept
Novice-4	It looks difficult
Novice-5	Typing (Textual language)
Intermediate-1	Mapping documents to nodes (Drag and drop would make it easier)
Intermediate-2	Mentally keeping where I am in the network
Intermediate-3	Mapping
Intermediate-4	Draw the flow with xTed
Intermediate-5	Make component and structure
Intermediate-6	No answer
Expert-1	Getting the concept of Petri net
Expert-2	No answer
Expert-3	Time consuming
Expert-4	Creating links in manual HTML creation

Table E.8: Answers for Question 20

Question 20	What is the least difficult task to perform?
Novice-1	Laying out the format
Novice-2	Library net and composing
Novice-3	Create hypertext
Novice-4	Drawing
Novice-5	Composing
Intermediate-1	N/A
Intermediate-2	Link the file to the place node
Intermediate-3	Authoring language
Intermediate-4	Language acceptable
Intermediate-5	Search
Intermediate-6	No answer
Expert-1	Design hypertext using the provided patterns
Expert-2	A unit structure can be generated in one time.
Expert-3	Automatic layout by clicking with library
Expert-4	Grouping multiple nets into one choice

Table E.9: Answers for Question 21

Question 21	What features of TcAT make it difficult to create hypertext?
Novice-1	It doesn't show the instant view of the web page.
Novice-2	TcAT didn't perform instant demonstration. Users might want to see how it works in real.
Novice-3	Difficult to define concept
Novice-4	It looks difficult
Novice-5	File name remembering (when mapping place to content file)
Intermediate-1	N/A
Intermediate-2	Fast growing # of branches due to Petri net's inherited property
Intermediate-3	It looks more complex when create a simple page
Intermediate-4	No answer
Intermediate-5	Needs to manage for long document
Intermediate-6	No answer
Expert-1	Bottom-up design method
Expert-2	No answer
Expert-3	Bug
Expert-4	No answer

Table E.10: Answers for Question 22

Question 22	What features of TcAT are most helpful in creating hypertext?
Novice-1	I don't need to know HTML to make a web page.
Novice-2	It shows to users whole picture of hypertext, and it helps.
Novice-3	Easy to create hypertext even though I didn't know how to create it.
Novice-4	I can see the whole picture.
Novice-5	Building blocks
Intermediate-1	Template and authoring language
Intermediate-2	Tree structure visualization, modulization, text editor (language)
Intermediate-3	It looks better when control complex and large amount of data
Intermediate-4	Suggest structure and expanded
Intermediate-5	Search and combine
Intermediate-6	No answer
Expert-1	template
Expert-2	We can see the outlined (overall) structure before providing the web page -> allow user to modify
Expert-3	Component in library
Expert-4	Graphic interface makes it easy

Table E.11: Answers for Question 23

Question 23	What features would you like to add to the TcAT?
Novice-1	Instant pre-view of the web page
Novice-2	Instant web page demonstration
Novice-3	Instant preview of web page
Novice-4	Preview
Novice-5	Drag and drop
Intermediate-1	Drag and drop and more functions in authoring language
Intermediate-2	N/A
Intermediate-3	Focusing
Intermediate-4	Drag and drop, check the related information in the same level of process by the file name
Intermediate-5	Backup (Undo function)
Intermediate-6	No answer
Expert-1	Detailed design of a hypertext maybe TcAT like web design tool
Expert-2	No answer
Expert-3	Drag and drop
Expert-4	Don’t have type net ID

Table E.12: Answers for Question 24

Question 24	What features would you like to remove from the TcAT?
Novice-1	None
Novice-2	No answer
Novice-3	None
Novice-4	Nothing
Novice-5	Nothing
Intermediate-1	No answer
Intermediate-2	Nothing specific
Intermediate-3	Nothing
Intermediate-4	Nothing
Intermediate-5	Nothing (good)
Intermediate-6	No answer
Expert-1	Nothing
Expert-2	No answer
Expert-3	N/A
Expert-4	No answer

VITA

Yung Ah Park was born in Bonghwa, South Korea. She received her B.S. degree in computer science from POSTECH, Pohang, South Korea in 1992, and earned her M.S. degree in computer science from Oklahoma State University, Stillwater, in 1997. She received a Ph.D. degree in computer science at Texas A&M University in August 2010, under the supervision of Dr. Richard Furuta. She worked in the Department of Computer Science and Engineering as a graduate assistant from January 2001 through December 2009. Her current research interests include hypertext/hypermedia, digital libraries, CSCW, HCI, user interfaces and information visualizations.

Ms. Park may be reached at Department of Computer Science and Engineering, Texas A&M University, TAMU 3112, College Station, TX 77843. Her email is pyungah@gmail.com.